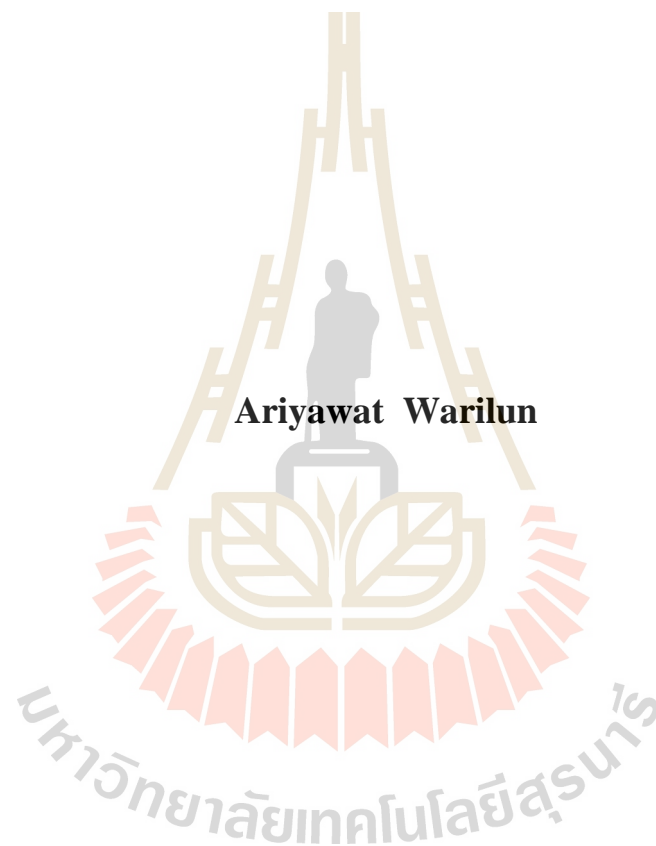


การออกแบบอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดสำหรับโรบอท  
โดยใช้การมองเห็นของเครื่องจักร



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมการผลิต  
มหาวิทยาลัยเทคโนโลยีสุรนารี  
ปีการศึกษา 2559

**DESIGN OF SHORTEST TRAJECTORY ALGORITHM  
FOR ROBOT WITH MACHINE VISION SYSTEM**



**A Thesis Submitted in Partial Fulfillment of the Requirement for the  
Degree of Master of Engineering in Manufacturing Engineering**

**Suranaree University of Technology**

**Academic Year 2016**

การออกแบบอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดสำหรับโรบอทโดยใช้การ  
มองเห็นของเครื่องจักร

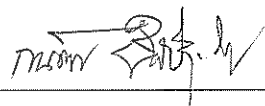
มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นักศึกษานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์




(อ. ดร.วรรณวนษ์ บุ่งสุต)

ประธานกรรมการ



(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



(ผศ. ดร.กัญชลา สุตตาชาติ)

กรรมการ



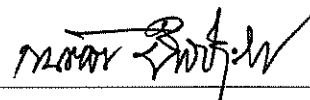
(อ. ดร.โสรัฏฐา แจ่มการ )

กรรมการ



(ศ. ดร.ชูกิจ ลิมปิจำนงค์)

รองอธิการบดีฝ่ายวิชาการและนวัตกรรม



(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)

คณบดีสำนักวิชาวิศวกรรมศาสตร์

อริยวัตร วริณ : การออกแบบอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดสำหรับโรบอทโดยใช้การมองเห็นของเครื่องจักร (DESIGN OF SHORTEST TRAJECTORY ALGORITHM FOR ROBOT WITH MACHINE VISION SYSTEM) อาจารย์ที่ปรึกษา :  
รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์, 109 หน้า.

งานวิจัยนี้เป็นการออกแบบอัลกอริทึมในการหาเส้นทางที่สั้นที่สุดของการเคลื่อนที่โดยใช้การแก้ปัญหาของพนักงานขายสำหรับการเคลื่อนที่ไปยังตำแหน่งต่างๆของโรบอท ซึ่งควบคุมโดยใช้การมองเห็นของเครื่องจักร ทั้งนี้เนื่องจากการทำงานของโรบอทสำหรับจ่ายของเหลวที่มีการจ่ายของเหลวหลายตำแหน่งจำเป็นต้องมีการหาเส้นทางที่สั้นที่สุดและมีการสอนโรบอทให้เคลื่อนที่ไปยังตำแหน่งต่างๆนอกจากนี้จะต้องมีการจัดวางตำแหน่งของผลิตภัณฑ์ให้ถูกต้องตามตำแหน่งที่ได้มีการทดสอบ หากตำแหน่งของผลิตภัณฑ์มีความคลาดเคลื่อนในลักษณะของการหมุนเพียงเล็กน้อยก็จะทำให้การจ่ายของเหลวไม่ตรงจุดที่ต้องการอาจทำให้ผลิตภัณฑ์ไม่สามารถนำไปใช้งานต่อได้งานวิจัยนี้คาดว่าจะทำให้การออกแบบกระบวนการผลิตมีความยืดหยุ่นมากขึ้นเพราะตำแหน่งของการวางตัวหรือรูปแบบของผลิตภัณฑ์สามารถปรับเปลี่ยนได้เนื่องจากการมองเห็นของเครื่องจักรสามารถตรวจจับตำแหน่งและเคลื่อนที่ไปยังตำแหน่งที่ต้องการได้อย่างถูกต้องอีกทั้งยังไม่ต้องมีการสอนโรบอทใหม่เมื่อมีการปรับเปลี่ยนผลิตภัณฑ์เนื่องจากโรบอทสามารถหาเส้นทางที่สั้นที่สุดในการเคลื่อนที่ไปยังตำแหน่งต่างๆได้ด้วยตัวเอง

ARIYAWAT WARILUN : DESIGN OF SHORTEST TRAJECTORY  
ALGORITHM FOR ROBOT WITH MACHINE VISION SYSTEM.  
THESIS ADVISOR : ASSOC. PROF. FLT. LT. KONTORN  
CHAMNIPRASART, Ph.D., 109 PP.

MACHINE VISION/ROBOT/DISPENSING MACHINE/TRAVELLING  
SALESMAN PROBLEM

This research aims to build and design the algorithm base on travelling salesman problem for the shortest trajectory of the robot with equip with vision system. This will increase potential to do a correct work for dispensing machine. The regular dispensing machine must be search the shortest trajectory and teach the robot before use in the production line. However, all part and equipment must be align to a precision position and rotation. Since a little of alignment can cause a bad part for the next process. This research aims to introduce more flexible production line. Because the machine vision can locate positions and move to the positions correctly, moreover the robot can search the shortest trajectory by itself when change a product.

School of Manufacturing Engineering

Academic Year 2016

Student's Signature  Ariyawat Warilun

Advisor's Signature  Kontorn Chamniprasart

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี ผู้วิจัยขอขอบพระคุณบุคคลและกลุ่มบุคคลต่างๆที่ให้คำปรึกษา แนะนำ และช่วยเหลืออย่างดีมาโดยตลอดทั้งด้านวิชาการและการดำเนินงานวิจัยดังนี้

รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์ คณบดีสำนักวิชาวิศวกรรมศาสตร์ และยังเป็นທີ່ปรึกษาวิทยานิพนธ์ ที่ได้มอบโอกาสทางการศึกษาในระดับบัณฑิตศึกษา อีกทั้งความรู้ด้านวิชาการ คำแนะนำในการทำวิจัย จนผ่านอุปสรรคต่างๆมาได้ด้วยดี และมอบโอกาสดีๆเสมอมา

อาจารย์ ดร.สมศักดิ์ ศิวดำรงพงศ์ รองคณบดีฝ่ายบริหาร ผู้ซึ่งประสาทวิชาความรู้ ให้คำแนะนำทั้งทางด้านวิชาการ และการดำเนินชีวิต จนสามารถผ่านอุปสรรคต่างๆได้ด้วยดี

ดร.พิจิตรา เอื้องไฟโรจน์ ผู้ซึ่งให้คำแนะนำช่วยเหลือด้านภาษาอังกฤษ ในการเขียนบทความ ตลอดจนให้คำแนะนำสำหรับการเขียนบทความที่ดี และให้ความช่วยเหลือจนสามารถผ่านอุปสรรคต่างๆ

นายพันศักดิ์ เนระแก และนางสาวกฤษณา เนระแก ที่ได้ช่วยในการจัดหาอุปกรณ์สำหรับการสร้างเครื่องจักร อีกทั้งยังให้คำแนะนำการออกแบบทางกลสำหรับการสร้างเครื่องจักร

บุคลากรประจำสำนักวิชาวิศวกรรมการผลิต และบุคลากรประจำสำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ที่ได้อำนวยความสะดวกในการทำงานวิจัยทั้งด้านเอกสารข้อมูลต่างๆที่ใช้ในงานวิจัย

สุดท้ายนี้ผู้วิจัยขอขอบคุณอันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอขอบแก่บิดามารดาอันเป็นที่เคารพยิ่งและคณาจารย์ผู้ประสาทความรู้ตลอดจนทุกๆท่านที่ให้กำลังใจจนกระทั่งวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

อริยวัตร วาริณ

# สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) .....	ก
บทคัดย่อ (ภาษาอังกฤษ) .....	ข
กิตติกรรมประกาศ .....	ค
สารบัญ .....	ง
สารบัญตาราง .....	ช
สารบัญรูป .....	ฉ
บทที่	
1 บทนำ .....	1
1.1 ความเป็นมาและความสำคัญของปัญหา .....	1
1.2 วัตถุประสงค์ของการวิจัย .....	2
1.3 ขอบเขตงานวิจัย .....	2
1.4 สถานที่ทำวิจัย .....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	3
2 ปรีทรรสน์วรรณกรรมที่เกี่ยวข้อง .....	4
2.1 กล่าวนำ .....	4
2.2 ประวัติความเป็นมาและวรรณกรรมที่เกี่ยวข้อง .....	4
2.3 ความรู้ทั่วไปเกี่ยวกับโรบอท .....	5
2.3.1 ประวัติความเป็นมาของหุ่นยนต์ .....	5
2.3.2 โครงสร้างของหุ่นยนต์ .....	6
2.3.3 ส่วนประกอบของหุ่นยนต์ .....	7
2.3.4 ชนิดของหุ่นยนต์ .....	8
2.3.5 การหา DH Parameters .....	11
2.3.6 การหาเวกเตอร์การเคลื่อนที่ของหุ่นยนต์ .....	12

## สารบัญ (ต่อ)

## หน้า

2.4	ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์	15
2.5	ระบบนิวเมติกส์	17
2.5.1	อุปกรณ์พื้นฐานที่ใช้ลมอัดเป็นต้นกำลัง	18
2.5.2	วาล์วควบคุมในระบบนิวเมติกส์	28
2.6	พื้นฐานระบบปรับภาพและระบบการสอบเทียบภาพ	31
2.6.1	ระบบรับภาพ	31
2.6.2	ความบิดเบี้ยวจากระบบรับภาพ	34
2.6.3	การสอบเทียบระบบปรับภาพเชิงพื้นที่	35
2.7	พื้นฐานภาพดิจิทัล	36
2.7.1	พื้นฐานสามประการของภาพดิจิทัล	36
2.7.2	ชนิดของภาพดิจิทัล	37
2.7.3	ประเภทของไฟล์ภาพ	38
2.7.4	รูปแบบการเก็บภาพในหน่วยความจำของ NI Vision	39
2.8	พื้นฐานการประมวลผลภาพ	40
2.8.1	ขอบภาพ	40
2.8.2	หน้ากากของภาพ (Image Mask)	43
2.9	ปัญหาการเดินทางของพนักงานขาย	45
2.9.1	หลักการของ Branch and Bound	46
2.9.2	หลักการของ 2-Opt	47
2.10	ความรู้ทั่วไปสำหรับภาษา C#	48
2.11	ความรู้ทั่วไปสำหรับ DLL	49
3	วิธีดำเนินงานวิจัย	50
3.1	กล่าวนำ	50
3.2	อุปกรณ์ที่สำคัญสำหรับการทำวิจัย	50



## สารบัญ (ต่อ)

### หน้า

3.2.1 Cartesian Robot	50
3.2.2 กล้องควบคุม	51
3.2.3 กระบอกลูกสูบ	52
3.2.4 Power Supply	53
3.2.5 Solenoid Valve	54
3.2.6 Filter Regulator	54
3.2.7 กล้อง Webcam	55
3.2.8 Microcontroller	56
3.2.9 Computer Laptop	58
3.2.10 Jig สำหรับจับปากกา	58
3.2.11 Reed Switch	59
3.3 การออกแบบอัลกอริทึม	60
3.3.1 ระบบการมองเห็นของเครื่องจักร	61
3.3.2 การควบคุม Cartesian Robot	61
3.3.3 การหาเส้นทางที่สั้นที่สุดโดยใช้ TSP	62
3.4 ขั้นตอนการทำงานของระบบ	62
3.4.1 การถ่ายภาพกริดสำหรับการปรับปรุงความผิดพลาดของภาพ	69
3.4.2 การบันทึกตำแหน่งอ้างอิงของโรบอทกับภาพที่ได้จากกล้อง	70
3.4.3 นำเข้าภาพกริดและตำแหน่งอ้างอิง	71
3.4.4 การนำเข้าภาพและปรับปรุงคุณภาพของภาพ	71
3.4.5 ทำการปรับความผิดพลาดของภาพ	72
3.4.6 การหาวงกลมภายในภาพ	73
3.4.7 การหาเส้นทางที่สั้นที่สุด	74
3.4.8 ทำการแปลงระบบโคออร์ดิเนต	79
3.4.9 ควบคุมโรบอทให้เคลื่อนที่ไปยังตำแหน่งต่างๆ	80

## สารบัญ (ต่อ)

หน้า

<b>4 ผลการทำงานของเครื่องจักรและอัลกอริทึม</b>	84
4.1 ผลการปรับปรุงความผิดเพี้ยนของภาพ	85
4.2 ผลการตรวจหาตำแหน่งของการเคลื่อนที่	86
4.3 ผลการคำนวณความเร็วในการหาเส้นทางในการเคลื่อนที่ไปยังตำแหน่งต่างๆ	87
4.4 ผลการปรับระบบโคออร์ดิเนตระหว่างกล้องกับโรบอท	89
4.5 ผลความแม่นยำในการเคลื่อนที่ไปยังตำแหน่งต่างๆ	89
4.6 ผลการทดสอบการเปรียบเทียบอัลกอริทึม	91
4.7 สรุปผลการทดสอบ	94
<b>5 สรุปผลและข้อเสนอแนะ</b>	93
5.1 สรุปผลการวิจัย	93
5.1.1 อัลกอริทึมสำหรับการหาเส้นทางที่สั้นที่สุด	93
5.1.2 การประมวลผลภาพและการควบคุมโรบอท	93
5.2 ข้อเสนอแนะ	94
รายการอ้างอิง	95
ภาคผนวก	
ภาคผนวก ก. รายละเอียดโปรแกรมย่อย Sub VI ที่ใช้ในงานวิจัย	97
ประวัติผู้เขียน	109

## สารบัญตาราง

ตารางที่	หน้า
2.1 รายละเอียดโครงสร้างของกระบอกสูบชนิดทางานทิศทางเดียว.....	20
2.2 แสดงรายละเอียดโครงสร้างของกระบอกสูบชนิดทางานสองทิศทาง.....	22
2.3 สัญลักษณ์ของจนวนตาแหน่งทางานของวาล์ว.....	29
2.4 ความหมายของสัญลักษณ์ช่องต่อท่อลม.....	29
2.5 สัญลักษณ์เส้นทางการไหลผ่านภายในวาล์วที่แสดงด้วยลูกศรและช่องปิด.....	30
2.6 ระยะเวลาระหว่างจุดของการเดินทาง.....	46
4.1 เวลาที่ใช้ในการประมวลผลเฉลี่ยในการหาเส้นทางที่สั้นที่สุด.....	87
4.2 ความคลาดเคลื่อนในแต่ละรูปแบบของตำแหน่งสำหรับการเคลื่อนที่.....	89



## สารบัญรูป

รูปที่

หน้า

2.1 ตัวอย่าง The Cartesian Robot	8
2.2 ตัวอย่าง The Cylindrical Robot	8
2.3 ตัวอย่าง The Spherical Robot	9
2.4 ตัวอย่าง The Articulated Robot หรือ The Revolute Robot	10
2.5 ตัวอย่าง The SCARA Robot	10
2.6 ตัวอย่าง The Parallel Robot	11
2.7 แสดงรายละเอียดของตำแหน่งตัวแปรต่างๆ ของ DH Parameters	12
2.8 แสดงตัวอย่างแบบจำลองของหุ่นยนต์ 2 แขน	13
2.9 วงจรระบบนิวเมติกส์พื้นฐาน	17
2.10 ภาพถ่ายอุปกรณ์ในระบบนิวเมติกส์จากวงจรนิวเมติกส์รูปที่ 2.9	18
2.11 องค์ประกอบภายในและหลักการทางานของกระบอกสูบทางเดียว	19
2.12 การสั่งงานให้กระบอกสูบทางเดียว	19
2.13 โครงสร้างภายในของกระบอกสูบทางเดียว	19
2.14 สัญลักษณ์ของกระบอกสูบทางเดียวแบบปกติเข้าและปกติออก	20
2.15 หลักการทำงานภายในของกระบอกสูบสองทางเมื่อจ่ายลมอัดภายใน	21
2.16 โครงสร้างภายในของกระบอกสูบชนิดทำงานสองทิศทาง	21
2.17 กระบอกสูบชนิดทำงานสองทางที่มีอุปกรณ์ป้องกันการกระแทก	22
2.18 ตัวอย่างกระบอกสูบชนิดต่างๆ	23
2.19 กระบอกสูบชนิดพิเศษและสัญลักษณ์กระบอกสูบชนิดต่างๆ	24
2.20 กระบอกสูบโรตารีมอเตอร์ลมและสัญลักษณ์	25
2.21 ตัวอย่างอุปกรณ์หีบจับจากยี่ห้อ Festo	26
2.22 หลักการทำงานภายในและการจับยึดชิ้นงานรูปทรงต่างๆ	27
2.23 ตัวอย่างการประยุกต์อุปกรณ์หีบจับที่ใช้งานร่วมกับหุ่นยนต์อุตสาหกรรม	28
2.24 สัญลักษณ์และการเรียกชื่อวาล์วชนิดต่างๆ	30
2.25 สัญลักษณ์ของรูปแบบที่ใช้ในการควบคุมวาล์ว	31
2.26 ขอบเขตที่กล้องรับภาพได้ หรือ FOV	33

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.27 ระยะชัดลึกของเลนส์	34
2.28 ระนาบที่บรรจุภาพสี	37
2.29 รูปแบบการเก็บภาพในหน่วยความจำของ NI Vision	40
2.30 วิธีการกำหนดค่าพิกเซลที่อยู่ในขอบด้วยวิธีต่าง ๆ	42
2.31 หลักการทำงานของการจัดส่วนแบ่งภาพ	43
2.32 การกำหนดค่าเอียงให้กับหน้าฉากของภาพ	44
2.33 ผลกระทบของการกำหนดค่าเอียงให้กับหน้าฉากของภาพ	45
2.34 การตัดเส้นทาง และเชื่อมเส้นทางใหม่ของ 2-Opt	48
3.1 Cartesian Robot โดยใช้ Linear Actuator ของบริษัท IAI	51
3.2 กล่องควบคุมของบริษัท IAI รุ่น PSEL-C-2-42PI-42PI-NP-2-0	52
3.3 ระบายอกสูบของบริษัท CKD รุ่น LCS-1240	53
3.4 Power supply ของบริษัท Omron	54
3.5 Solenoid Valve ของบริษัท SMC รุ่น VKF332-5HS-M5	54
3.6 Filter Regulator ของบริษัท SMC รุ่น AW10-M5BG	55
3.7 กล้อง webcam ที่ถูกติดตั้งไว้ส่วนบนของเครื่องจักร	56
3.8 IC เบอร์ ULN2803A	57
3.9 ไมโครคอนโทรลเลอร์ของบริษัท Arduino รุ่น Mega 2560	58
3.10 Computer laptop ของบริษัท Samsung รุ่น RF509-S01TH	58
3.11 Jig สำหรับจับปากกาขนาด 8 – 12 มิลลิเมตร	59
3.12 การติดตั้ง Reed Switch กับระบายอกสูบ	60
3.13 เครื่องสำหรับทดสอบอัลกอริทึม	62
3.14 การเชื่อมต่อระหว่างอุปกรณ์	64
3.15 ขั้นตอนการทดสอบ	64
3.16 ขั้นตอนการทำงานของระบบ	65
3.17 ขั้นตอนการทำงานของระบบ (ต่อ)	66
3.18 ขั้นตอนการทำงานของระบบ (ต่อ)	67
3.19 ขั้นตอนการทำงานของระบบ (ต่อ)	68

## สารบัญรูป (ต่อ)

รูปที่	หน้า
3.20 ขั้นตอนการทำงานของระบบ (ต่อ)	69
3.21 ระยะระหว่างวงกลมเท่ากับ 10 มิลลิเมตร ตามแกน x และ แกน y	70
3.22 ภาพกริดที่ถ่ายจากกล้อง เพื่อใช้ในการปรับความผิดเพี้ยนของภาพ	70
3.23 เลือกไฟล์ภาพและตำแหน่งอ้างอิงเพื่อใช้ในการคำนวณ	71
3.24 Block diagram module การปรับความผิดเพี้ยนของภาพ	72
3.25 การทดสอบ module เพื่อใช้ในการปรับความผิดเพี้ยนของภาพและคำนวณ หาตำแหน่งของวงกลมภายในภาพรวมถึงคำนวณตำแหน่งในโลกจริง	73
3.26 Block diagram module สำหรับการหาวงกลมภายในภาพและการวาดรูปวงกลม สีน้ำเงินทับวงกลมที่ถูกลบภายในภาพ	74
3.27 ขั้นตอนการหาเส้นทางที่สุดโดยใช้ Branch and bound	76
3.28 ขั้นตอนการหาเส้นทางที่สุดโดยใช้ Branch and bound (ต่อ)	77
3.29 ขั้นตอนการปรับปรุงเส้นทางให้สั้นลงโดยใช้ 2-Opt โดยที่ n คือจำนวนจุดทั้งหมด	78
3.30 ขั้นตอนการปรับปรุงเส้นทางให้สั้นลงโดยใช้ 2-Opt โดยที่ n คือจำนวน จุดทั้งหมด (ต่อ)	79
3.31 ขั้นตอนการปรับปรุงเส้นทางให้สั้นลงโดยใช้ 2-Opt โดยที่ n คือจำนวน จุดทั้งหมด (ต่อ)	80
3.32 การควบคุมโรบอทไปยังตำแหน่งต่างๆ	81
3.33 การควบคุมโรบอทไปยังตำแหน่งต่างๆ (ต่อ)	82
3.34 การควบคุมโรบอทไปยังตำแหน่งต่างๆ (ต่อ)	83
4.1 เครื่องจักรต้นแบบที่ใช้ในการทดสอบอัลกอริทึม	84
4.2 รูปแบบของตำแหน่งในการเคลื่อนที่	85
4.3 ลักษณะความผิดเพี้ยนของภาพเมื่อทดสอบด้วย NI Vision Builder	86
4.4 ผลการหาตำแหน่งของวงกลมและวาดทับด้วยวงกลมสีน้ำเงิน	87
4.5 ตัวอย่างผลการหาเส้นทางที่สั้นที่สุดจากโปรแกรมที่สร้างขึ้น	88
4.6 กราฟความสัมพันธ์ระหว่างจำนวนจุดกับเวลาเฉลี่ยที่ใช้ในการประมวลผล	89
4.7 กราฟแจกแจงความถี่ของความคลาดเคลื่อนทั้งหมด 741 ตำแหน่ง	90
4.8 ลักษณะความคลาดเคลื่อนทั้ง 741 ตำแหน่ง	91

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.9 รูปแบบการเปรียบเทียบที่ 1 และตำแหน่งต่างๆ	92
4.10 รูปแบบการเปรียบเทียบที่ 2 และตำแหน่งต่างๆ	92
4.11 รูปแบบการเปรียบเทียบที่ 3 และตำแหน่งต่างๆ	93
4.12 รูปแบบการเปรียบเทียบที่ 4 และตำแหน่งต่างๆ	93
4.13 รูปแบบการเปรียบเทียบที่ 5 และตำแหน่งต่างๆ	94
ก.1 Sub VI Axis Status X-Y Command คำสั่งสอบถามสถานะและตำแหน่งปัจจุบัน ของโรบอท	99
ก.2 Sub VI Axis Status ทำการถอดรหัสสถานะปัจจุบันของโรบอท	99
ก.3 Sub VI Hex to Dec แปลงเลขฐาน 16 เป็นเลขฐาน 10	99
ก.4 Sub VI Home X-Y คำสั่งให้โรบอทหาตำแหน่งโฮม	99
ก.5 Sub VI On-Off Servo คำสั่งเปิดการทำงานของมอเตอร์	100
ก.6 Sub VI Robot move X-Y คำสั่งให้โรบอทเคลื่อนที่ในพิกัด X และพิกัด Y	102
ก.7 Sub VI Robot Reset คำสั่งสำหรับ Reset โรบอท	102
ก.8 Sub VI Send Receive data ใช้ในการรับและส่งข้อมูลระหว่างคอมพิวเตอร์ และโรบอท	102
ก.9 Sub VI Robot Status ทำการถอดรหัสสถานะและตำแหน่งปัจจุบันของโรบอท	103
ก.10 Sub VI Interface Arduino Read ทำการถอดรหัสข้อมูล Input จาก Arduino	103
ก.11 Sub VI Interface Arduino Write ส่งคำสั่งให้ Arduino ควบคุมกระบอกสูบ	104
ก.12 Sub VI Image Calibration การปรับปรุงความผิดเพี้ยนของภาพ	104
ก.13 Sub VI Calibration Axis การปรับเลื่อนตำแหน่งและการหมุนของแกนภาพ	104
ก.14 Sub VI Find Circle การหาตำแหน่งของวงกลมและวาดภาพทับวงกลมที่หาเจอ	105
ก.15 Sub VI Find Rectangle การหาตำแหน่งสี่เหลี่ยมและวาดภาพทับสี่เหลี่ยมที่หาเจอ	105
ก.16 Sub VI Kernel Filter การประมวลผลภาพด้วยหน้ากาก	106
ก.17 Sub VI Change zero index การเปลี่ยนตำแหน่งอะเรย์ตัวที่ศูนย์กับตำแหน่ง ที่อยู่ใกล้กับตำแหน่ง x เท่ากับศูนย์และ y เท่ากับศูนย์มากที่สุด	106

## สารบัญรูป (ต่อ)

รูปที่

หน้า

ก.18	Sub VI Salesman algorithm การหาเส้นทางที่สั้นที่สุดด้วย Branch and bound และ 2-Opt.....	107
ก.19	Sub VI Salesman algorithm การหาเส้นทางที่สั้นที่สุดด้วย Branch and bound และ 2-Opt (ต่อ).....	107
ก.20	Sub VI Salesman travel การคัดแยกข้อมูลสำหรับการประมวลผลและนำผลที่ได้จากการทำ Branch and bound และ 2-Opt ไปวาดภาพเพื่อแสดงเส้นทางที่หาได้.....	108





# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในอุตสาหกรรมการผลิตอุปกรณ์อิเล็กทรอนิกส์จำเป็นต้องมีการจ่ายของเหลวไม่ว่าจะเป็น กาว ตะกั่ว สารเรืองแสงสำหรับหลอดไฟ LED และอื่นๆ ในงานเหล่านี้จำเป็นต้องใช้ความละเอียด และระมัดระวังสำหรับการจ่ายของเหลว ซึ่งการใช้มนุษย์นั้นสามารถทำได้ยากและต้องใช้เวลาในการทำงาน เนื่องจากการจ่ายของเหลวในปริมาณที่น้อยมากๆ และต้องการความแม่นยำในการจ่ายของเหลวที่ค่อนข้างสูงนั้น ซึ่งต้องใช้ผู้ปฏิบัติการที่มีความสามารถสูง ดังนั้นจึงจำเป็นต้องใช้เครื่องจักรในการจ่ายของเหลว โดยใช้ระบบหัวจ่ายซึ่งมีการควบคุมด้วยแบบอิเล็กทรอนิกส์และติดตั้งเข้ากับหุ่นยนต์ เพื่อช่วยในการทำงานสำหรับการควบคุมปริมาณของเหลวให้ได้ปริมาณที่เหมาะสม การควบคุมตำแหน่งของการจ่ายของเหลว และความแม่นยำสำหรับการจ่ายของเหลวลงบนชิ้นงาน ปัจจุบันในกระบวนการผลิตมีการใช้หุ่นยนต์กันอย่างแพร่หลาย หน้าที่ของหุ่นยนต์คือการทำงานซ้ำๆติดต่อกันเป็นเวลานาน เพื่อลดความผิดพลาดที่อาจเกิดขึ้นจากความเมื่อยล้าของมนุษย์ การสั่งการเพื่อให้หุ่นยนต์ทำงานจำเป็นต้องมีการสอนและโปรแกรม เพื่อให้หุ่นยนต์จ่ายของเหลวลงบนชิ้นงานที่มีการจัดวางตำแหน่งและแนวการวางตัวของชิ้นงานที่แน่นอน เพื่อให้ระบบสามารถทำงานได้และเกิดความผิดพลาดน้อยที่สุด สำหรับกระบวนการนี้วิศวกรและช่างเทคนิคจำเป็นต้องออกแบบระบบจับยึดชิ้นงานให้มีความคลาดเคลื่อนน้อยที่สุด ทั้งตำแหน่งของชิ้นงานและแนวการวางตัวของชิ้นงาน เพราะความคลาดเคลื่อนที่เกิดขึ้นนั้นหมายถึงคุณภาพของชิ้นงาน และถ้าหากเกิดการเอียงเพียงเล็กน้อยของการวางชิ้นงาน จะทำให้ตำแหน่งของการจ่ายของเหลวผิดพลาด ซึ่งอาจทำให้ชิ้นงานเหล่านั้นใช้งานไม่ได้แน่นอนผลที่ตามมาก็คือต้นทุนและเวลาที่จะสูญเสียไป ดังนั้นการออกแบบจะต้องมีความรอบคอบ เพื่อใช้ในการผลิตชิ้นงานจำนวนมากและไม่จำเป็นต้องแก้ไขกระบวนการบ่อยครั้งเมื่อมีการเปลี่ยนผลิตภัณฑ์ เนื่องจากถ้ามีการเปลี่ยนผลิตภัณฑ์จะพบว่าวิศวกรและช่างเทคนิคจำเป็นต้องออกแบบระบบที่ใช้จับยึดชิ้นงานใหม่ มีการสอนและโปรแกรมหุ่นยนต์ขึ้นมาใหม่ ซึ่งระบบจำเป็นจะต้องมีการหยุดพัก หรือ

อาจต้องมีการสร้างเครื่องจักรใหม่สำหรับผลิตภัณฑ์ใหม่ๆ ที่มีการเปลี่ยนแปลงรูปร่างไปค่อนข้างมาก โดยระบบนี้จะเหมาะสมกับโรงงานที่มีจำนวนรุ่นของผลิตภัณฑ์ไม่มากนัก

อย่างไรก็ตามในปัจจุบันการพัฒนาและการเติบโตของเทคโนโลยีเป็นไปอย่างรวดเร็ว การเกิดของเทคโนโลยีใหม่ๆ และมีผลิตภัณฑ์ใหม่ๆ ออกมาเพื่อตอบสนองแก่ผู้บริโภคมากขึ้น มีรูปแบบของผลิตภัณฑ์ที่มากขึ้นระยะเวลาของการเปลี่ยนแปลงเทคโนโลยีเร็วขึ้น การผลิตผลิตภัณฑ์ต้องการความยืดหยุ่นในการผลิตสินค้ามากขึ้น ผู้ผลิตต้องปรับเปลี่ยนเครื่องจักรให้มีความสามารถมากขึ้น เพื่อให้เครื่องจักรสามารถทำงานได้กับหลายผลิตภัณฑ์ด้วยเครื่องจักรแบบเดียวกัน

งานวิจัยนี้เป็นการเสนอแนวแนวทางใหม่สำหรับกระบวนการหาตำแหน่งที่จะจ่ายของเหลว โดยจะทำการจัดสร้างเครื่องต้นแบบขึ้นมาแบบสำหรับกระบวนการหาตำแหน่งที่จะจ่ายของเหลวโดยหุ่นยนต์ ซึ่งในที่นี้จะเรียกแทนว่า โรบอท ซึ่งโรบอทจะทำการมองเห็นจากกล้องที่เราติดตั้งบนเครื่องจักรในการตรวจหาตำแหน่งที่จะทำการจ่ายของเหลวและใช้คอมพิวเตอร์ในการคำนวณหาเส้นทางที่สั้นที่สุดสำหรับการจ่ายของเหลวในแต่ละจุด และสั่งการให้โรบอทเคลื่อนที่ไปยังตำแหน่งต่างๆ ได้อย่างถูกต้อง

## 1.2 วัตถุประสงค์ของการวิจัย

สำหรับงานวิจัยนี้ มีวัตถุประสงค์ของงานวิจัยในด้านการออกแบบอัลกอริทึมโดยการใช้การมองเห็นของเครื่องจักรและคอมพิวเตอร์ ในการหาตำแหน่งสำหรับการเคลื่อนที่ของโรบอท และเคลื่อนที่ไปตามเงื่อนไขที่กำหนด ซึ่งมีวัตถุประสงค์หลักดังนี้

1.2.1 จัดสร้างเครื่องต้นแบบที่ใช้การมองเห็นของเครื่องจักรในการหาตำแหน่งการเคลื่อนที่และควบคุมโรบอทให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการได้อย่างถูกต้อง

1.2.2 ออกแบบอัลกอริทึมสำหรับการเคลื่อนที่ของโรบอท โดยสามารถหาเส้นทางที่สั้นที่สุดสำหรับการเดินทางของโรบอทได้

## 1.3 ขอบเขตของงานวิจัย

ขอบเขตการวิจัยนี้เพื่อสามารถกำหนดเป้าหมายและวิธีการแก้ปัญหาได้อย่างเป็นระบบ โดยที่ขอบเขตของงานวิจัยนี้ประกอบด้วย

1.3.1 สร้างเครื่องต้นแบบสำหรับหาตำแหน่งในการเคลื่อนที่ให้สามารถทำงานตามที่กำหนด โดยที่เครื่องต้นแบบที่สร้างขึ้นเป็นเพียงเครื่องต้นแบบในระดับปฏิบัติการเท่านั้น

1.3.2 รูปแบบของผลิตภัณฑ์และเครื่องหาตำแหน่งของการเคลื่อนที่ถูกจำลองขึ้นเพื่อการทดลอง

1.3.3 โรบอทที่ใช้เป็นแบบ Cartesian Robot จำนวน 1 ชุด

1.3.4 ใช้การมองเห็นของเครื่องจักรในการหาตำแหน่งสำหรับการเคลื่อนที่

1.3.4 ใช้การแก้ปัญหาแบบ Travelling Salesman Problem (TSP) ในการหาเส้นทางที่สั้นที่สุดโดยใช้อัลกอริทึมที่เหมาะสม

1.3.5 เครื่องต้นแบบสามารถเคลื่อนที่ไปยังตำแหน่งที่ต้องการได้อย่างถูกต้อง

#### 1.4 สถานที่ทำงานวิจัย

1.4.1 อาคารเครื่องมือวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีสุรนารี

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ได้แนวคิดใหม่สำหรับการใช้ระบบตรวจจับตำแหน่งและการเคลื่อนที่ไปยังตำแหน่งต่างๆ

1.5.2 ช่วยให้อาจารย์วางแผนกระบวนการผลิตได้ง่ายขึ้น

1.5.3 ได้แนวทางสำหรับการพัฒนาโรบอทให้มีความฉลาดมากขึ้น

1.5.4 ได้แนวคิดสำหรับกระบวนการผลิตที่มีความยืดหยุ่นมากขึ้น

มหาวิทยาลัยเทคโนโลยีสุรนารี

## บทที่ 2

### ปริทรรศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง

#### 2.1 กล่าวนำ

ในบทนี้จะกล่าวถึงปริทรรศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้องกับงานวิจัยฉบับนี้ และยังกล่าวถึงความเป็นมาของการพัฒนาเทคโนโลยีด้านการมองเห็นของเครื่องจักรอุตสาหกรรมที่มีการค้นคว้าวิจัยในด้านนี้มาเป็นเวลากว่า 60 ปี ซึ่งการพัฒนาการมองเห็นของเครื่องจักรนั้นเริ่มต้นพัฒนาเพื่อใช้ในการทหาร ก่อนจะแพร่หลายเข้าสู่ภาคอุตสาหกรรม ประกอบกับอุปกรณ์ที่ใช้ในระบบการมองเห็นของเครื่องจักรมีราคาถูกลงและสามารถซื้อหาได้ง่ายขึ้น งานวิจัยด้านการมองเห็นของเครื่องจักรจึงได้รับการวิจัยอย่างต่อเนื่องและแพร่หลาย

ในส่วนแรกนี้เป็นส่วนที่ผู้วิจัยได้ทำการค้นคว้าเอกสารงานวิจัยต่างๆ ที่เกี่ยวข้องกับงานวิจัยที่ได้ดำเนินการอยู่ และตามที่ได้กล่าวมาแล้วข้างต้น เนื่องจากงานวิจัยด้านนี้มีการวิจัยมาอย่างยาวนานมีบทความและเอกสารที่เกี่ยวข้องเป็นจำนวนมาก ประกอบทั้งการประมวลผลภาพด้วยคอมพิวเตอร์มีงานวิจัยเปลี่ยนแปลงในหลากหลายด้านและพัฒนาอย่างไปอย่างรวดเร็ว ดังนั้นผู้วิจัยได้เลือกเฉพาะงานวิจัยที่เกี่ยวข้องหรือคล้ายคลึงกับงานวิจัยที่กำลังดำเนินการอยู่เท่านั้น ในส่วนของบทนี้ผู้วิจัยจะนำเสนอทฤษฎีที่เกี่ยวข้องในงานวิจัยได้แก่ ความรู้ทั่วไปของโรบอท ความรู้เบื้องต้นสำหรับไมโครคอนโทรลเลอร์ การมองเห็นของเครื่องจักร และการหาเส้นทางที่สั้นที่สุดโดยการประยุกต์ตัวแบบสำหรับการแก้ปัญหาการเดินทางของพนักงานขาย (Travelling Salesman Problem: TSP)

#### 2.2 ประวัติความเป็นมาและปริทรรศน์วรรณกรรมที่เกี่ยวข้อง

การมองเห็นของเครื่องจักรเป็นงานวิศวกรรมด้านระบบเชิงกลไฟฟ้า การจับภาพด้วยกล้องและการใช้หรือการพัฒนาซอฟต์แวร์ให้ระบบต่างๆ ให้สามารถทำงานร่วมกันได้ เพื่อใช้ตรวจสอบความสมบูรณ์ของผลิตภัณฑ์ที่ได้มาจากระบบการผลิต หรือช่วยในการประกอบชิ้นส่วน ตลอดจนใช้ในการควบคุมการทำงานของเครื่องจักร จากยุค ค.ศ. 1980 จนถึงปัจจุบัน ได้มีการคิดวิธีการประมวลผลภาพและวิเคราะห์ภาพเพื่อใช้ในงานอุตสาหกรรมเกิดการประมวลผลภาพหลายทฤษฎี โดยมีการพยายามเชื่อมโยงการประมวลผลภาพเข้ากับกลไกหรือโรบอทเพื่อให้เกิดเครื่องจักรที่มีความฉลาด คือสามารถนำภาพที่ตรวจจับมาใช้เพื่อควบคุมการตัดสินใจ สำหรับการมองเห็นของ

เครื่องจักรใน อุตสาหกรรมของประเทศไทยจะเน้นไปที่กระบวนการตรวจสอบในหลายอุตสาหกรรม โดยการประยุกต์การใช้งานก็จะแตกต่างกัน อย่างไรก็ตามการพัฒนาซอฟต์แวร์นั้นจะถูกพัฒนาขึ้นเพื่อผลิตภัณฑ์เพียงแบบเดียว และการกำหนดรูปแบบของการเคลื่อนที่ของโรบอทนั้นยังต้องอาศัยการสอนตำแหน่งให้กับโรบอทและกำหนดเส้นทางการเคลื่อนที่

ในส่วนของการประมวลผลภาพนั้นผู้วิจัยใช้โปรแกรมสำเร็จรูปคือ LabVIEW ของบริษัท Nation Instrument (NI) และโปรแกรม LabVIEW เองก็จะมี module สำหรับการประมวลผลภาพคือ NI Vision สำหรับการประมวลผลภาพ ทางด้านการหาทางเดินที่สั้นที่สุดนั้นผู้วิจัยใช้ ภาษา C# โดยใช้ตัวแบบสำหรับการแก้ปัญหาการเดินทางของพนักงานขาย (TSP) เพื่อเขียน library สำหรับโปรแกรม LabVIEW

ปัญหาการเดินทางของพนักงานขาย หรือ TSP นั้นเป็นที่สนใจของนักวิจัยจำนวนมาก เนื่องจาก TSP เป็นปัญหาที่แก้ได้ยากจัดอยู่ในปัญหาแบบ NP-hard และปัจจุบันก็ยังไม่สามารถแก้ปัญหได้ด้วยวิธีการแบบ polynomial ได้ สำหรับช่วงเวลาที่ผ่านมาผู้วิจัยได้ทำการค้นคว้าและพบเห็นงานวิจัยที่เกี่ยวข้องกับงานวิจัยที่ได้ทำอยู่ดังนี้

Richard Wiener (2003) [3, 4] ได้นำเสนอการใช้ Branch and Bound ในการแก้ปัญหาการเดินทางของพนักงานขาย ซึ่งเป็นกระบวนการแก้ปัญหาที่สามารถแก้ปัญหา การเดินทางของพนักงานขายได้อย่างรวดเร็ว

Justine Jackson, et al (2007) [5] ได้นำเสนอการแก้ปัญหา TSP โดยใช้ Tabu search ในการหาเส้นทางเริ่มต้นและปรับปรุงเส้นทางให้สั้นลงโดยใช้ 2-Opt

Handa A. Darwish and Ihab Talkhan (2014) [6] ได้นำเสนอการลดความซับซ้อนของ การแบ่งแยกเคสเอาชนะ (Divide and conquer) ในการแก้ปัญหา TSP โดยการ optimize ด้วย 2-opt, 3-opt และ k-opt

## 2.3 ความรู้ทั่วไปเกี่ยวกับโรบอท

### 2.3.1 ประวัติความเป็นมาของหุ่นยนต์

หุ่นยนต์เป็นเทคโนโลยีที่ใช้กันอย่างแพร่หลายในอุตสาหกรรมการผลิตเพราะความสามารถที่ทำงานแทนมนุษย์ได้ สามารถทำงานบางอย่างได้ดีกว่าที่มนุษย์สามารถทำได้และเกิดความผิดพลาดน้อยกว่าเมื่อทำงานเป็นเวลานานปัจจุบันหุ่นยนต์ถูกพัฒนาและออกแบบมาให้มีความสามารถเพิ่มมากขึ้นเพื่อตอบสนองต่อความต้องการต่ออุตสาหกรรมการผลิต เช่น สามารถโปรแกรมให้ทำตามคำสั่งได้มากขึ้น มีความเร็วในการเคลื่อนที่ไปยังตำแหน่งต่างๆได้เร็วขึ้น มีความแม่นยำมากขึ้น เป็นต้น หุ่นยนต์หรือที่เรียกกันว่าโรบอทนั้นเริ่มต้นเมื่อปี ค.ศ. 1760 ได้มีการ

นำเอาเครื่องจักรไอน้ำเข้ามาทำงานแทนที่คนงานผลที่ได้คือเครื่องจักรทำงานได้ดีกว่ามนุษย์จากนั้นเรื่องก็เงียบมาจนถึงปี ค.ศ. 1920 ได้มีนักเขียนละครชาวเชกโกสโลวาเกียชื่อ Karel Capek ได้เขียนบทละครเรื่อง Rossum's Universal Robots : R.U.R. โดยคำว่า Robot เป็นคำแผลงมาจากคำว่า Robotla ในภาษาเช็กที่แปลว่า Worker หรือคนงานต่อมาในปี ค.ศ. 1942 ได้นักประพันธ์ทางด้านวิทยาศาสตร์ Isaac Asimov ได้บัญญัติกฎของหุ่นยนต์ขึ้น 3 ข้อคือ

- ข้อหนึ่ง หุ่นยนต์ห้ามทำร้ายมนุษย์หรือปล่อยให้มนุษย์ได้รับอันตราย
- ข้อสอง หุ่นยนต์ต้องเชื่อฟังคำสั่งของมนุษย์เว้นแต่จะขัดแย้งกับกฎข้อหนึ่ง
- ข้อสาม หุ่นยนต์ต้องป้องกันตัวเองโดยไม่ขัดแย้งกับกฎข้อหนึ่งและข้อสอง

ต่อมาภายหลัง ได้มีการเพิ่มกฎข้อศูนย์ลงไปคือ

- กฎข้อศูนย์ หุ่นยนต์มีอำนาจกระทำอันตรายต่อมนุษยชาติ หรือนิ่งเฉยปล่อยให้มนุษยชาติตกอยู่ในอันตรายได้โดยการการำตามกฎข้อ 1, 2 และ 3 จะต้องไม่ขัดกับข้อที่ 0

ถัดมาในปี ค.ศ. 1954 ได้มีการสร้างหุ่นยนต์ตัวแรกของโลกโดย George C. Devol วิศวกรชาวอเมริกาพร้อมกับจดสิทธิบัตรและจากนั้นก็ได้มีการพัฒนาเทคโนโลยีหุ่นยนต์เรื่อยมาจนถึงปัจจุบัน

### 2.3.2 โครงสร้างของหุ่นยนต์

โครงสร้างของหุ่นยนต์สามารถแบ่งออกเป็น 2 ส่วนสำคัญคือ ส่วนลำตัว-แขน (Body - Arm) และข้อมือ (Wrist) โดยมากแล้วส่วนลำตัวและแขนจะมีระดับขั้นความอิสระ 3 ระดับ ในส่วนของข้อมือจะมีระดับขั้นของความอิสระอยู่ 2 – 3 ระดับ ที่ส่วนปลายของข้อมือจะเป็นวัตถุซึ่งมีความสัมพันธ์กับงานที่หุ่นยนต์ต้องทำตัวอย่างเช่น อาจเป็นชิ้นงานที่ต้องการโหลดเข้าเครื่อง หรืออาจเป็นเครื่องมือที่หุ่นยนต์ต้องใช้ในการผลิตบางอย่าง เป็นต้น ส่วนลำดับของแขนและขาของหุ่นยนต์จะใช้ในการจัดตำแหน่งที่ถูกต้องของชิ้นงาน ส่วนของข้อมือจะใช้สำหรับการจัดทิศทางและการวางตัวของชิ้นงาน เพื่อที่จะวางตำแหน่งของวัตถุให้เหมาะสมส่วนลำตัวและแขนของหุ่นยนต์จะต้องเคลื่อนย้ายชิ้นงานได้สามทิศทางดังต่อไปนี้

1. การเคลื่อนที่ในแนวตั้ง (การเคลื่อนที่ตามแนวแกน z)
2. การเคลื่อนที่ตามแนวรัศมี (การเคลื่อนที่เข้าหรือออกหรือการเคลื่อนที่ตามแนวแกน y)
3. การเคลื่อนที่จากซ้ายไปขวา (การเคลื่อนที่ในแนวแกน x)

การที่หุ่นยนต์จะสามารถเคลื่อนที่ได้ตามลักษณะที่กล่าวมาแล้วข้างต้นนี้ได้หลายวิธีทั้งนี้ขึ้นอยู่กับกรอบแบบและชนิดของข้อต่อที่ใช้ในการสร้างลำตัวและขาของหุ่นยนต์ซึ่งจะกล่าวรายละเอียดต่อไป เพื่อที่จะทำให้สามารถจัดทิศทางที่เหมาะสมของชิ้นงานเราสามารถกำหนด



ระดับชั้นความอิสระ 3 ระดับสำหรับข้อมือของหุ่นยนต์ ตัวอย่างต่อไปนี้เป็นหนึ่งในรูปแบบในการประกอบข้อมือของหุ่นยนต์เพื่อทำให้เกิด 3 ลำดับชั้นของความอิสระ (3 Degree of Freedom)

1. Roll ลำดับชั้นความอิสระลักษณะนี้ทำได้โดยใช้ข้อต่อแบบ T เพื่อที่จะหมุนชิ้นงานรอบแกนของแขน

2. Pitch การหมุนขึ้นลงของชิ้นงานซึ่งทำได้โดยใช้ข้อต่อแบบ R (Rotational Joint)

3. Yaw การหมุนซ้ายและขวาของชิ้นงานซึ่งทำได้โดยใช้ข้อต่อแบบ R (Rotational Joint)

### 2.3.3 ส่วนประกอบของหุ่นยนต์

**2.3.3.1 Actuator** เป็นอุปกรณ์หรือชุดขับเคลื่อนเพื่อให้แขนกลหรือหุ่นยนต์เคลื่อนไหว ได้แก่ มอเตอร์ กระจบบอกสูบ เป็นต้น

**2.3.3.2 Manipulator** เป็นส่วนประกอบเพื่อช่วยเรื่องของการเคลื่อนไหวก่อประกอบด้วยแขน (Links) และข้อต่อ (Joints)

**2.3.3.4 End Effector** เป็นส่วนสุดท้ายของหุ่นยนต์เพื่อใช้ในการทำงานต่างๆเช่น มือสำหรับจับสิ่งของ อุปกรณ์ที่ใช้ในทางเทคนิคประเภทงานกัด กลึง เชื่อม หรือพ่น เป็นต้น

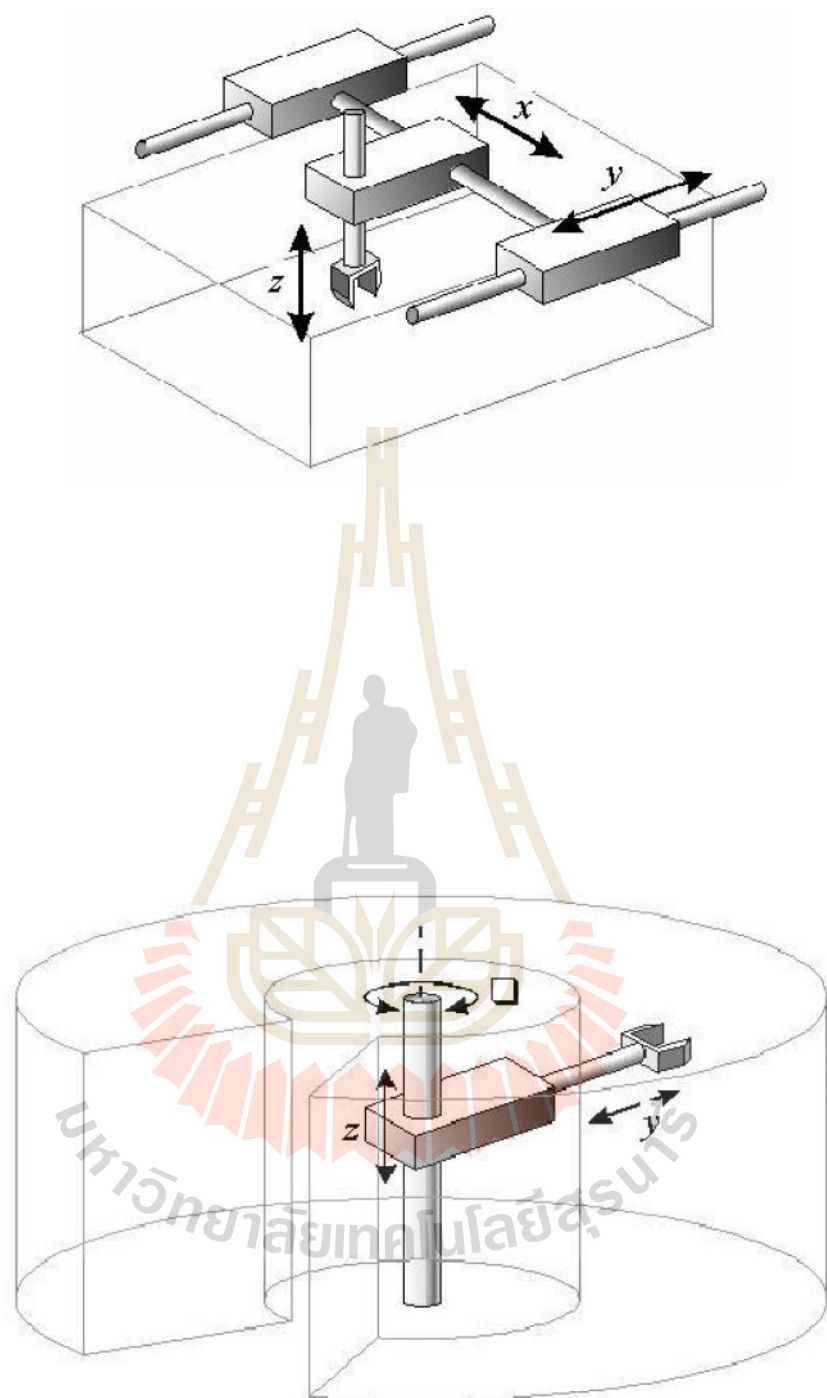
**2.3.3.5 Power Supply** เป็นแหล่งจ่ายพลังงานสำหรับอุปกรณ์ต่างๆของแขนกล เช่น กระแสไฟฟ้า หรือ ลมโดยใช้วาล์วในการควบคุมกระจบบอกสูบ

**2.3.3.6 Controller** เปรียบเหมือนสมองที่ใช้ในการสั่งการและควบคุมขั้นตอนต่างๆในการทำงานของแขนกล โดยส่งไปยังชุดขับเคลื่อนเพื่อให้เคลื่อนที่ตามที่ได้ตั้งโปรแกรมไว้

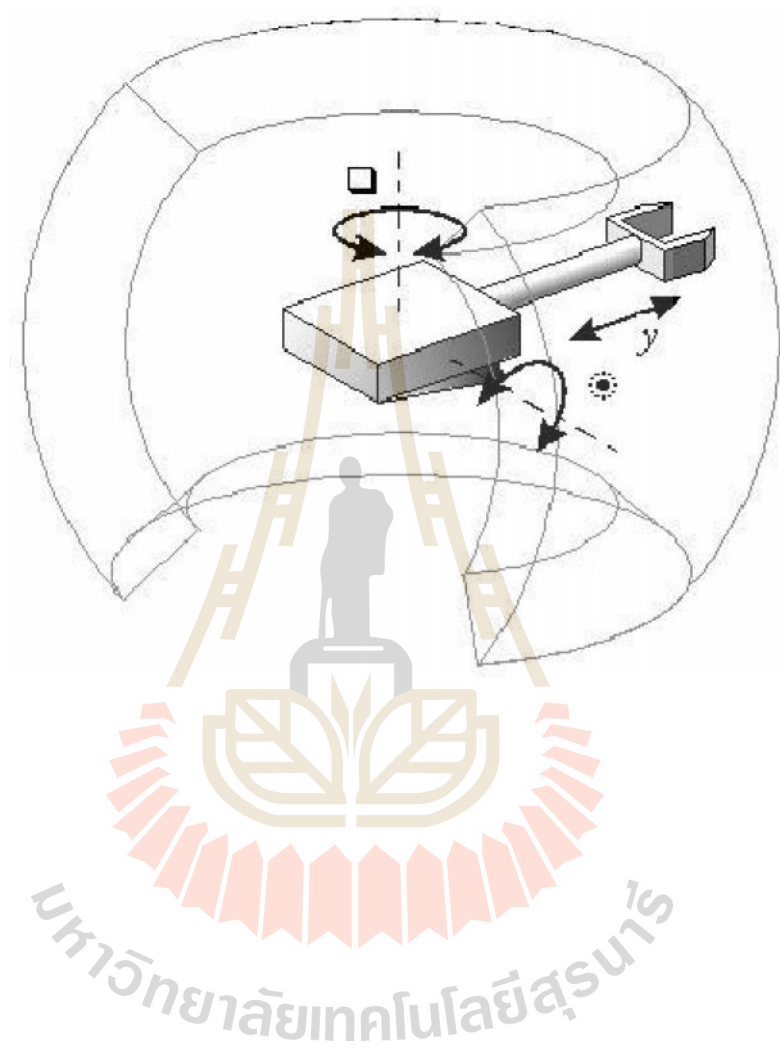
**2.3.3.7 Program** คือการกำหนดขั้นตอนต่างๆ การควบคุมตำแหน่ง รวมไปถึงจนถึงความเร็ว เพื่อให้โรบอททำงานได้ตามที่ต้องการ

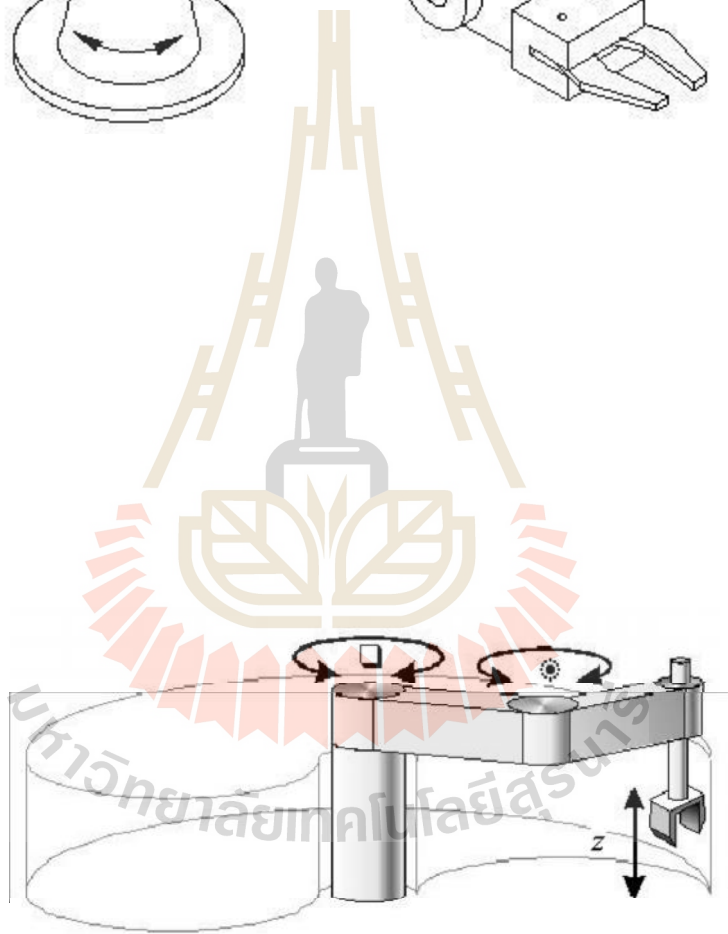
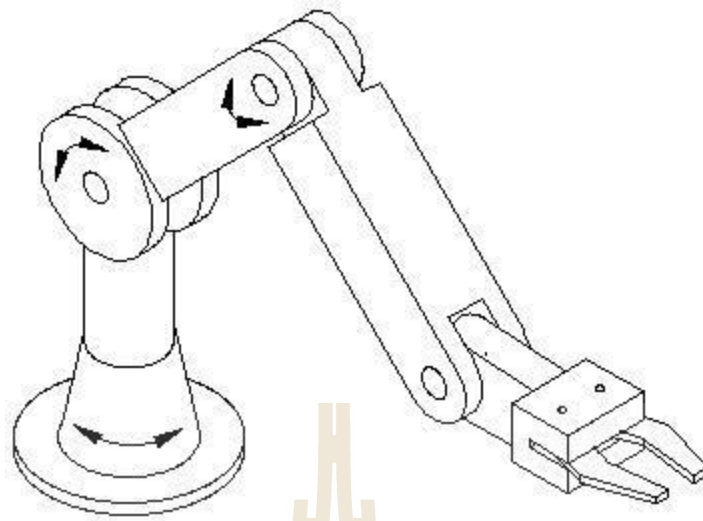
### 2.3.4 ชนิดของหุ่นยนต์

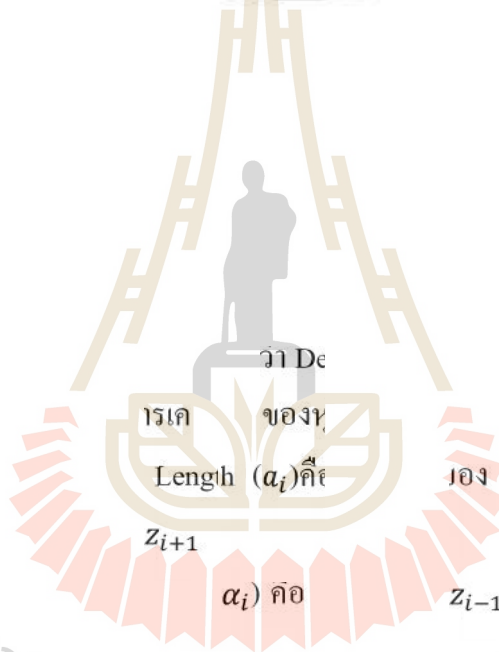
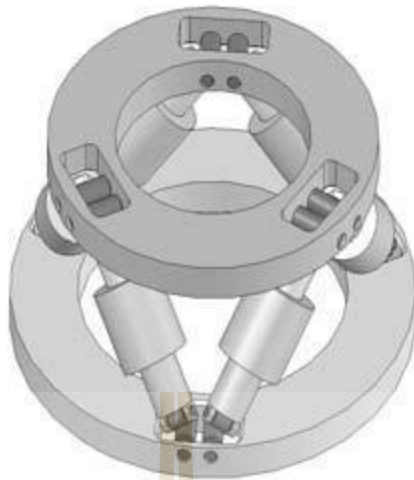
**2.3.4.1 The Cartesian Robot** เป็นหุ่นยนต์ที่มีการเคลื่อนที่แบบเชิงเส้นทั้งสามแกน มีลักษณะการทำงานที่ไม่ซับซ้อน สามารถทำความเข้าใจได้ง่าย แต่หุ่นยนต์ชนิดนี้มีข้อจำกัดในเรื่องของพื้นที่ที่สามารถทำงานได้ เพราะอาณาเขตของการเคลื่อนที่เป็นแบบกล่องสี่เหลี่ยม ส่วนมากหุ่นยนต์ชนิดนี้จะใช้ในการเคลื่อนย้ายของหนัก ซึ่งหุ่นยนต์ประเภทนี้จะมีลักษณะดังภาพประกอบ 2.1











มหาวิทยาลัยเทคโนโลยีสุรนารี

ว่า De

rg P

างคั'

ารเค

ปรที่

ของ

อง

$z_i$  ถึ

Length ( $a_i$ ) คื

$x_i$  ที่ตั้ง

$z_{i+1}$

$z_{i-1}$

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

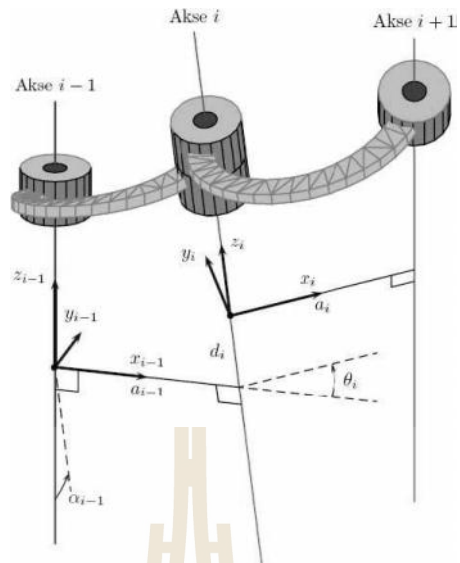
$z_i$  ที่มอง

$x_{i-1}$

$\alpha_i$  คื

$z_i$  ที่มอง

$x_{i-1}$



$$T_{i-1}^i = \begin{vmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$T_{i-1}^i$

$i$

$a_i$

$\theta_i$

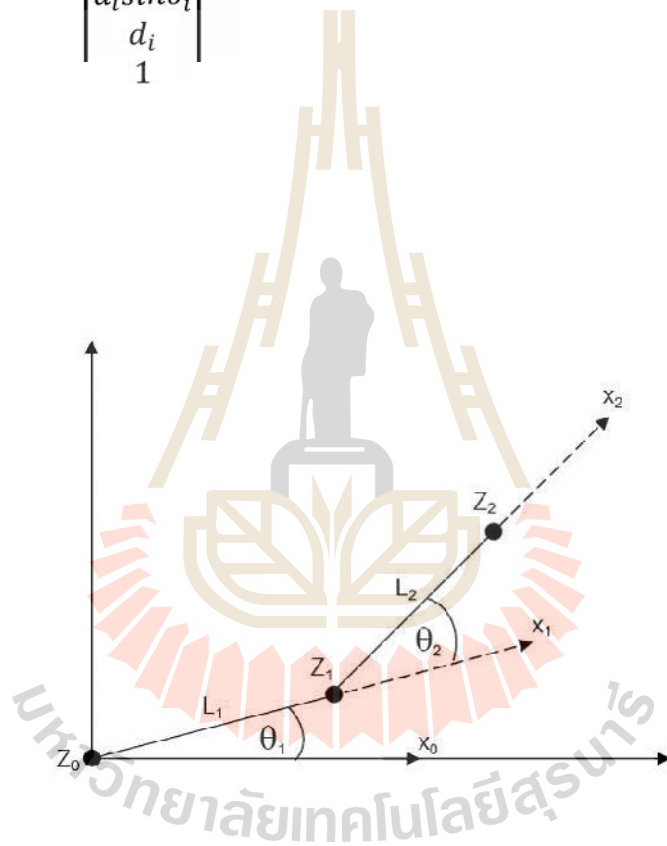
$d_i$

$\alpha_i$

## 2.1 จะ เคมตรกษการหมุน และเวกเตอรการเค

$$\begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix}$$

$$\begin{bmatrix} a_i\cos\theta_i \\ a_i\sin\theta_i \\ d_i \\ 1 \end{bmatrix}$$



.....

$a_i$   $L_1$

$\theta_i$   $\theta_1(t)$

$d_i$

$\alpha_i$

แขน  $L_2$  มี DH Parameters 4 ตัว ดังนี้

$a_i$  มีค่าเป็น  $L_2$

$\theta_i$  มีค่าเป็น  $\theta_2(t)$

$d_i$  มีค่าเป็น 0

$\alpha_i$  มีค่าเป็น 0

หลังจากหาค่า DH Parameters เสร็จแล้ว ต่อไปเป็นการคำนวณหาทรานส์ฟอร์มเมชันเมทริกซ์ (Transformation Matrix) ได้ดังนี้

$$T_0^1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & L_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & L_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$T_1^2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$T_0^2 = \begin{bmatrix} \cos(\theta_1 - \theta_2) & -\sin(\theta_1 - \theta_2) & 0 & L_1\cos(\theta_1 - \theta_2) \\ \sin(\theta_1 - \theta_2) & \cos(\theta_1 - \theta_2) & 0 & L_1\sin(\theta_1 - \theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

จากสมการที่ 2.6 จะได้เวกเตอร์การเคลื่อนที่ (Translation vector) ของหุ่นยนต์ตามแนวพิกัด xyz ได้ดังนี้

1. แกน x มีเวกเตอร์การเคลื่อนที่คือ  $L_1\cos\theta + L_2\cos(\theta_1 + \theta_2)$
2. แกน y มีเวกเตอร์การเคลื่อนที่คือ  $L_1\sin\theta + L_2\sin(\theta_1 + \theta_2)$
3. แกน z มีเวกเตอร์การเคลื่อนที่คือ 0 (ไม่มีการเคลื่อนที่ในแนวแกน z)

เท่านี้จะได้สมการต่างๆ ของเวกเตอร์การเคลื่อนที่บนแกนต่างๆของหุ่นยนต์

## 2.4 ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) มาจากคำสองคำคือ ไมโคร (Micro) หมายถึงขนาดเล็ก และคำว่า คอนโทรลเลอร์ (Controller) หมายถึงตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์จึงหมายถึง อุปกรณ์ควบคุมซึ่งมีขนาดเล็ก แต่ตัวอุปกรณ์ควบคุมขนาดเล็กนี้ได้บรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ ที่คนโดยส่วนใหญ่คุ้นเคยกล่าวคือ ภายในไมโครคอนโทรลเลอร์ ได้รวมเอาซีพียู หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวเดียวกัน

นักออกแบบพัฒนาผลิตภัณฑ์ ตลอดจนนักประดิษฐ์ทั้งหลายต่างๆ หลีกเลียงไม่ได้เลยที่จะต้องอาศัยวงจรอิเล็กทรอนิกส์เข้าไปมีส่วนเกี่ยวข้องในการควบคุม แต่ครั้งวงจรอิเล็กทรอนิกส์ที่นำมาต่ออนุกรมเพื่อเพิ่มความสามารถให้เป็นไปตามที่เราต้องการนั้นก็ใหญ่โตเสียเหลือเกิน ดูเหมือนจะขัดแย้งกับความต้องการของผู้บริโภค และหลักการออกแบบผลิตภัณฑ์ ดังนั้นไมโครคอนโทรลเลอร์จึงเข้ามาเกี่ยวข้องเพื่อรองรับกับความต้องการ เพื่อนำไปควบคุมระบบให้มีความสามารถตามที่เราต้องการ โดยให้มีขนาดเล็กที่สุด แต่มีใช้เพียงขนาดเล็กเท่านั้น มันยังสามารถป้อนชุดคำสั่งให้สามารถทำงานได้อย่างอัตโนมัติ ด้วยรูปแบบการเขียนโปรแกรม ภาษาต่าง ๆ ตามความถนัด

โครงสร้างโดยทั่วไป ของไมโครคอนโทรลเลอร์นั้นสามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

- หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)
- หน่วยความจำ (Memory) ซึ่งสามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้

สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง

- อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) เป็นเหมือนกระดาษทดในการ

คำนวณของซีพียู นอกจากนั้นยังเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่ถ้าหากไม่มีไฟเลี้ยงข้อมูลก็จะหายไป คล้ายกับหน่วยความจำ (RAM) ในเครื่องคอมพิวเตอร์ทั่ว ๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรมซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

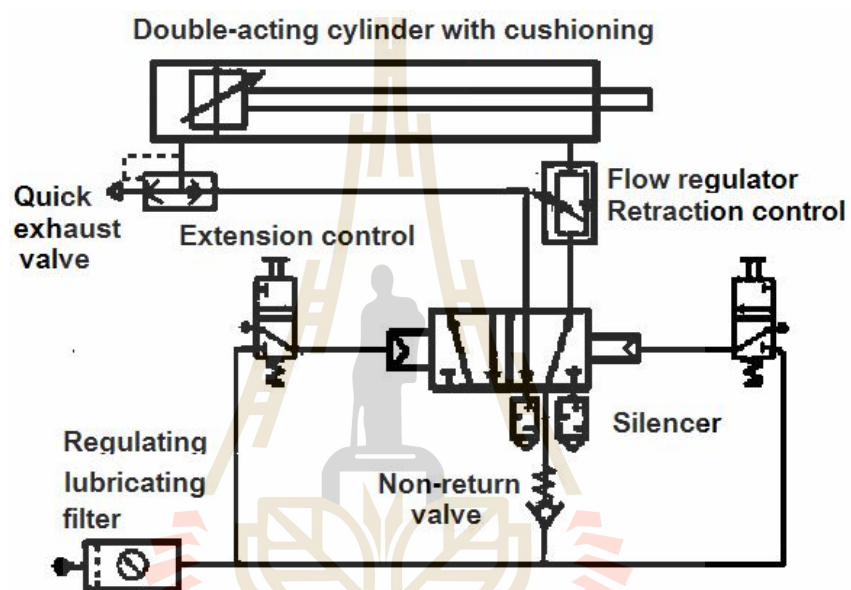
- ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต

(Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมากเมื่อใช้ร่วมกันระหว่างพอร์ตอินพุตเพื่อรับสัญญาณ อาจจะใช้วิธีการกดสวิตช์เพื่อนำไปประมวลผล และส่งไปพอร์ตเอาต์พุตเพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

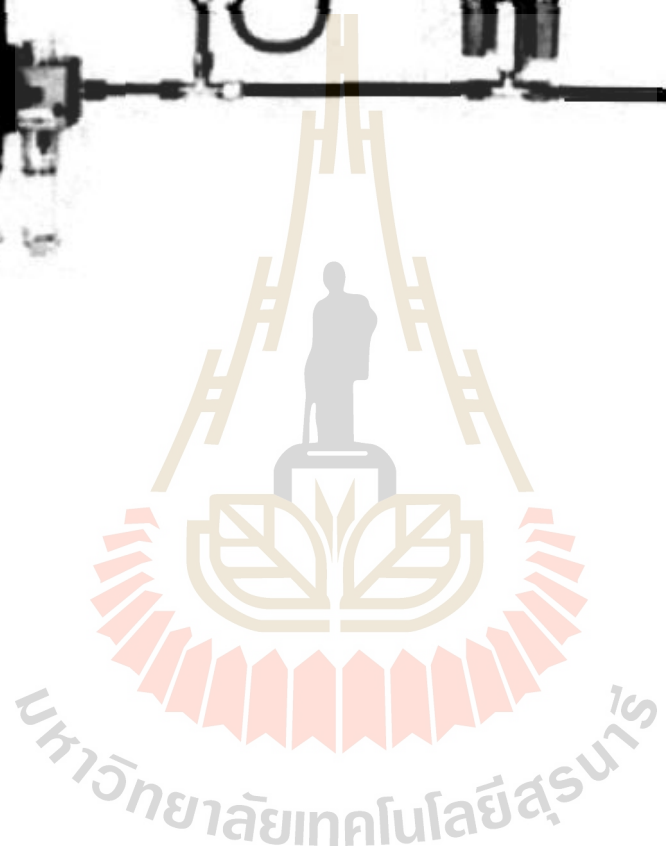
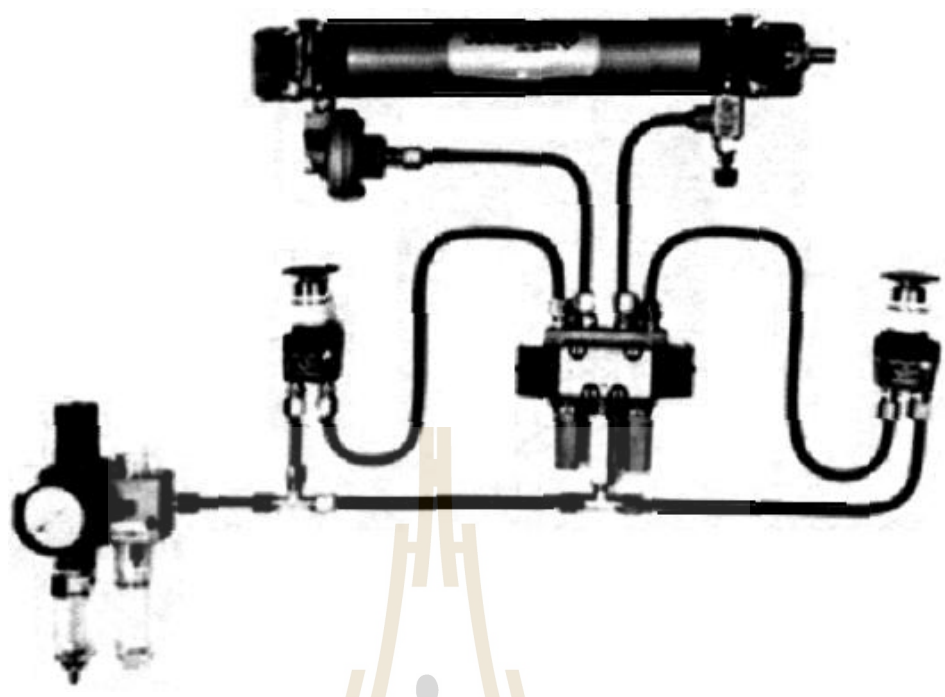
- ช่องทางเดินของสัญญาณหรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูล

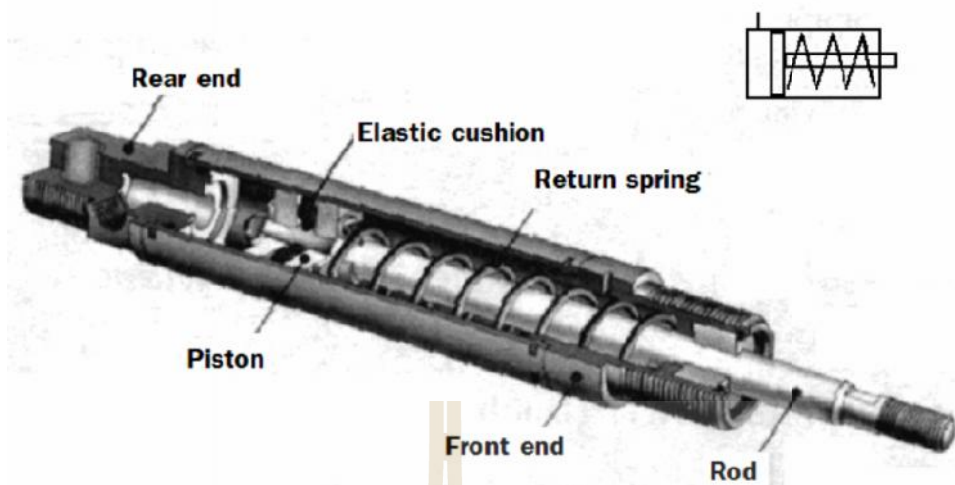
ระหว่าง ซีพียู หน่วยความจำ และพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ในตัวไมโครคอนโทรลเลอร์โดยแบ่งเป็นบัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus) บัสข้อมูลเป็นสายสัญญาณที่บรรจุข้อมูลเพื่อการประมวลผลทั้งหมด ขนาดของบัสจะขึ้นอยู่กับความสามารถในการประมวลผลของซีพียูสำหรับในงานทั่ว ๆ ไป ขนาดของบัสข้อมูลจะเป็น 8 บิต และในปัจจุบันได้มีการพัฒนาขึ้นมาจนถึง 16,32 และ 64 บิต บัสแอดเดรสเป็นสายสัญญาณที่บรรจุค่าตำแหน่งของหน่วยความจำ โดยการติดต่อกับหน่วยความจำนั้น ซีพียูต้องกำหนดตำแหน่งที่ต้องการอ่านหรือเขียนก่อน ดังนั้นจำนวนสายสัญญาณของแอดเดรสจึงต้องมีจำนวนมาก ยิ่งมากเท่าไร ก็จะเป็นการแสดงขนาดของหน่วยความจำที่ไมโครคอนโทรลเลอร์สามารถติดต่อได้ โดยสามารถคำนวณได้จากจำนวนแอดเดรสของหน่วยความจำ  $= 2^n$  ( $n$  คือจำนวนของเส้นทาง) ยกตัวอย่างเช่น ไมโครคอนโทรลเลอร์ตัวหนึ่งมีสายแอดเดรส 10 เส้น ดังนั้นไมโครคอนโทรลเลอร์ตัวนี้สามารถติดต่อกับหน่วยความจำได้ 2 ยกกำลัง 10  $= 1,024$  ตำแหน่งหากต้องการทราบความจุของหน่วยความจำจริงๆ จะต้องทราบถึงขนาดของบัสข้อมูลก่อนว่าเป็นเท่าใด หากเป็น 8 บิตความจุของหน่วยความจำที่มีสายแอดเดรส 10 เส้น จะเท่ากับ  $8 \times 1,024 = 8,192$  บิต และ 1 กิโลไบต์ เท่ากับ 1,024 ไบต์ ดังนั้นไมโครคอนโทรลเลอร์ดังกล่าวจึงมีความจุของหน่วยความจำเท่ากับ 8,192 บิต หรือ 1,024 ไบต์ หรือ 1 กิโลไบต์ บัสควบคุมเป็นกลุ่มของสายสัญญาณควบคุมการติดต่อทั้งหมดของซีพียูกับหน่วยความจำและพอร์ต สำหรับสายสัญญาณจะเลือกควบคุมหลักได้แก่ สายสัญญาณเลือก-อ่าน-เขียน หน่วยความจำสายสัญญาณเลือกจะเลือกอ่าน-เขียน ข้อมูล กับพอร์ตวงจรกิจการสัญญาณที่มีนาฬิกาเป็นส่วนใหญ่เป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่งเนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับกำหนัดจังหวะโดยถ้าหากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้น ส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้นมีความเร็วในการประมวลผลสูงตามไปด้วย



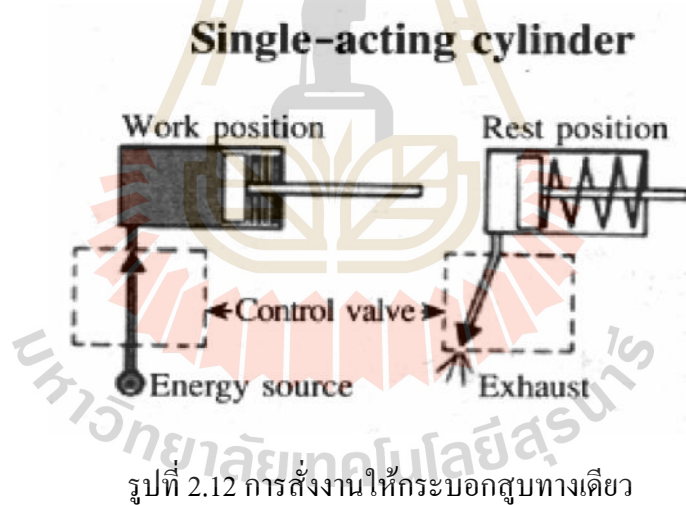


มหาวิทยาลัยเทคโนโลยีสุรนารี

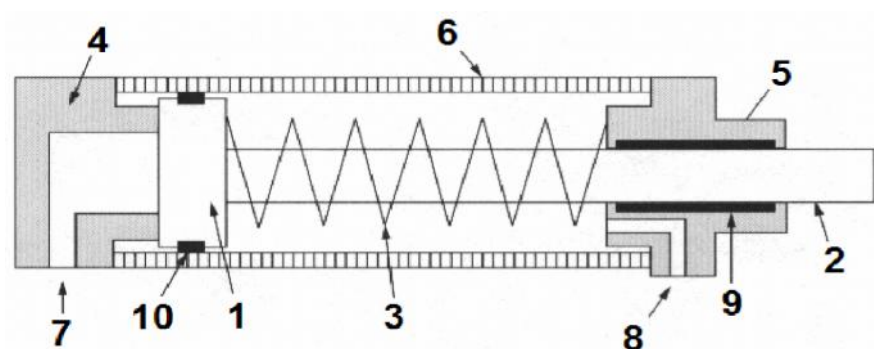




รูปที่ 2.11 องค์ประกอบภายในและหลักการทำงานของกระบอกสูบทางเดียว



รูปที่ 2.12 การสั่งงานให้กระบอกสูบทางเดียว



รูปที่ 2.13 โครงสร้างภายในของกระบอกสูบทางเดียว (รายละเอียดดังตารางที่ 2.1)

หมายเลข

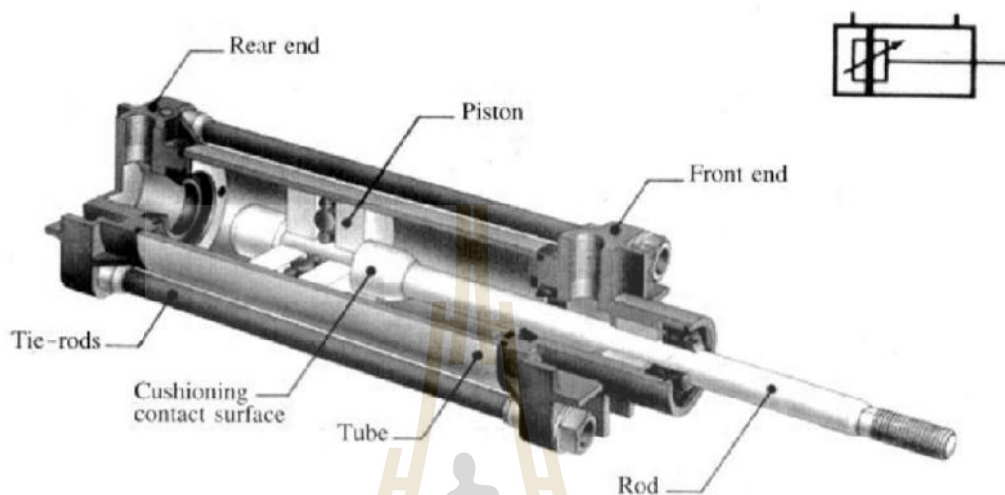
รายละเอียด

- |      |                                       |
|------|---------------------------------------|
| 1    | ลูกสูบ (piston)                       |
| 2    | ก้านสูบ (piston rod)                  |
| 3    | สปริงดันกลับ Return spring            |
| 4    | ฝาครอบท้าย (base end cover)           |
| 5    | ฝาครอบหัว (head end cover)            |
| 6    | กระบอกสูบ (cylinder tube)             |
| 7    | รูต่อลม (pressure connector )         |
| 8    | บุชก้านสูบ (bush and sealing element) |
| 9,10 | ซีลลูกสูบ (piston seal)               |

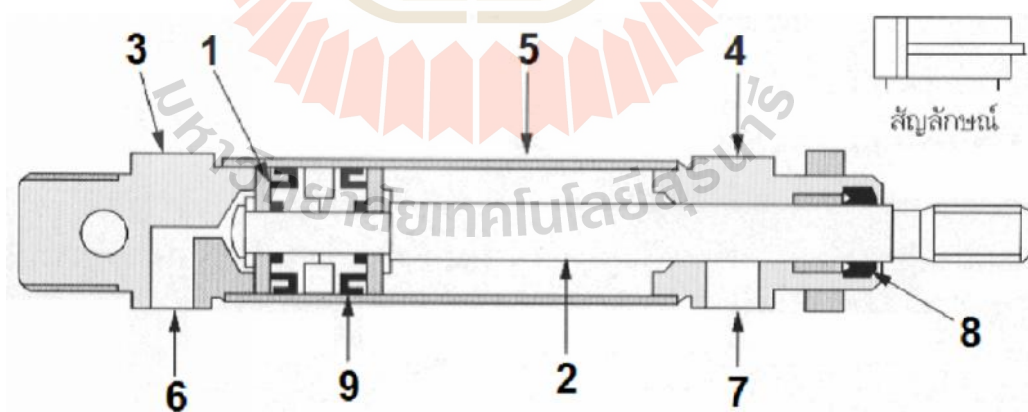


มหาวิทยาลัยเทคโนโลยีสุรนารี

ก้านสูบได้จะทำให้สามารถเคลื่อนไป-มาได้โดยอิสระ การแสดงส่วนประกอบภายในกระบอบอกสูบสองทางเป็นไปดังรูปที่ 2.16 และตารางที่ 2.2 แสดงชื่อส่วนประกอบภายในกระบอบอกสูบ



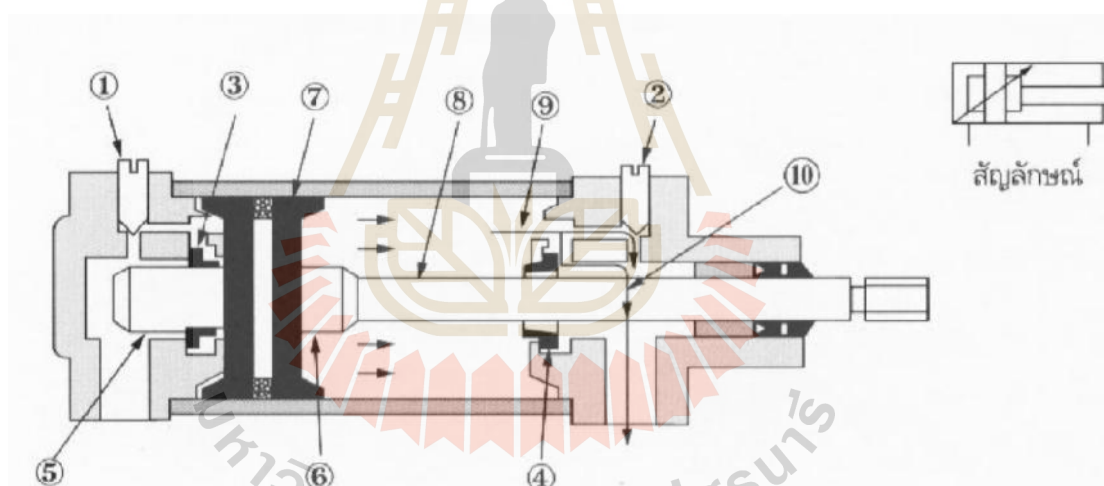
รูปที่ 2.15 หลักการทำงานภายในของกระบอบอกสูบสองทางเมื่อจ่ายลมอัดภายใน



รูปที่ 2.16 โครงสร้างภายในของกระบอบอกสูบชนิดทำงานสองทิศทาง

ตารางที่ 2.2 แสดงรายละเอียดโครงสร้างของกระบอกสูบชนิดทางานสองทิศทาง

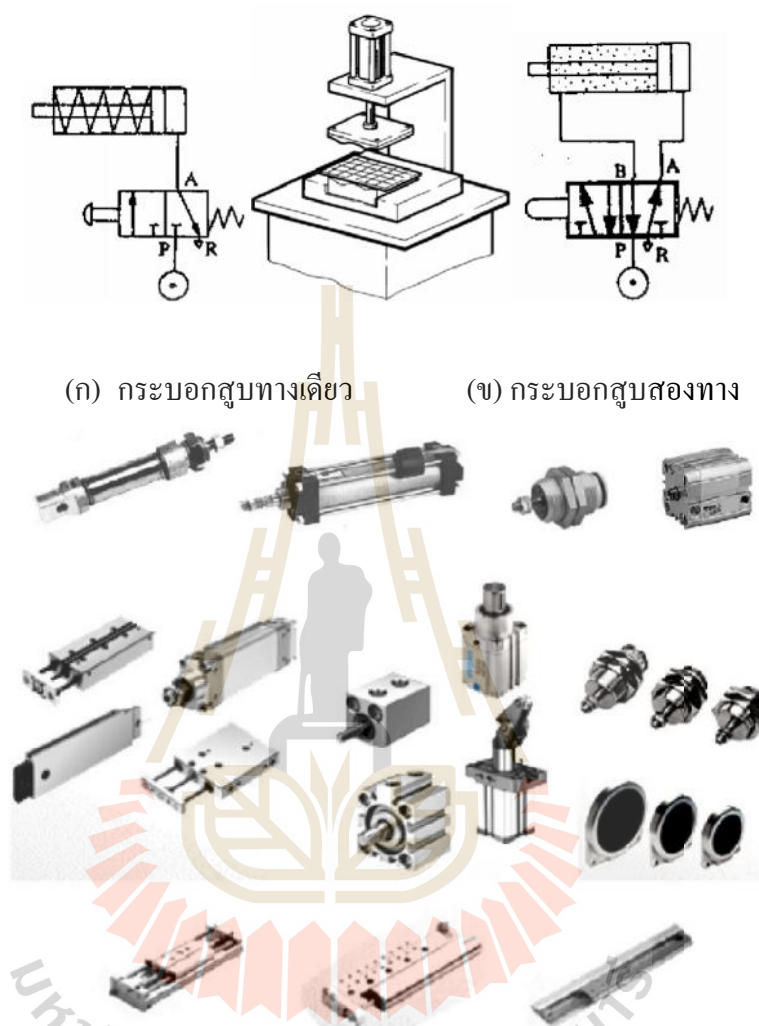
หมายเลข	รายละเอียด
1	ลูกสูบ (piston)
2	ก้านสูบ (piston rod)
3	ฝาครอบท้าย (base end cover)
4	ฝาครอบหัว (head end cover)
5	กระบอกสูบ (cylinder tube)
6	รูต่อลมด้านลูกสูบ (pressure connector , base side)
7	รูต่อลมด้านก้านสูบ (pressure connector, head side)
8	ซีลก้านสูบ (bush and sealing element)
9	ซีลลูกสูบ (piston seal)



รูปที่ 2.17 กระบอกสูบชนิดทางานสองทางที่มีอุปกรณ์ป้องกันการกระแทก

จากรูปที่ 2.17 เป็นกระบอกสูบสองทางที่มีอุปกรณ์กันการกระแทก เพื่อป้องกันความเสียหายจากการชนของก้านสูบกับกระบอกสูบ เมื่อก้านสูบเคลื่อนที่ด้วยความเร็วสูง หรือเมื่อมีการใช้งานลมอัดความดันสูง โดยหลักการในการลดความเร็วของก้านสูบมีดังนี้คือ ปกติลมอัดภายในกระบอกสูบจะระบายออกทางเส้นทางหมายเลข 9 และ 10 โดยสะดวก แต่เมื่อเดือย (6) เคลื่อนที่มาดันซีล (4) จะปิดทางลมหมายเลข 10 ทำให้ความเร็วของก้านสูบก่อนการกระแทกจะลดลง เนื่องจากลมจะระบายออกจากกระบอกสูบได้เฉพาะเส้นทางหมายเลข 9 ซึ่งสามารถปรับ

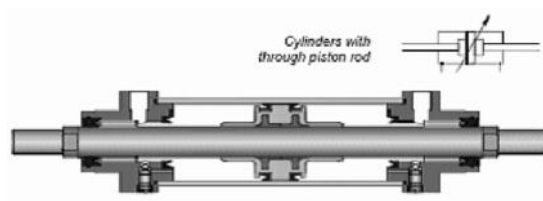
อัตราไหลของเส้นทางหมายเลข 9 ได้จากการปรับวาล์วลีลม (2) ทำให้เกิดแรงดันจากลมอัดที่ค้างอยู่ภายในกระบอกที่ไม่สามารถระบายออกอย่างรวดเร็วได้



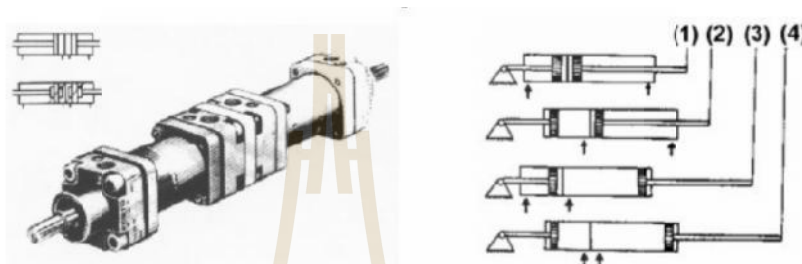
รูปที่ 2.18 ตัวอย่างกระบอกสูบชนิดต่างๆ

### 2.5.1.3 กระบอกสูบชนิดพิเศษ

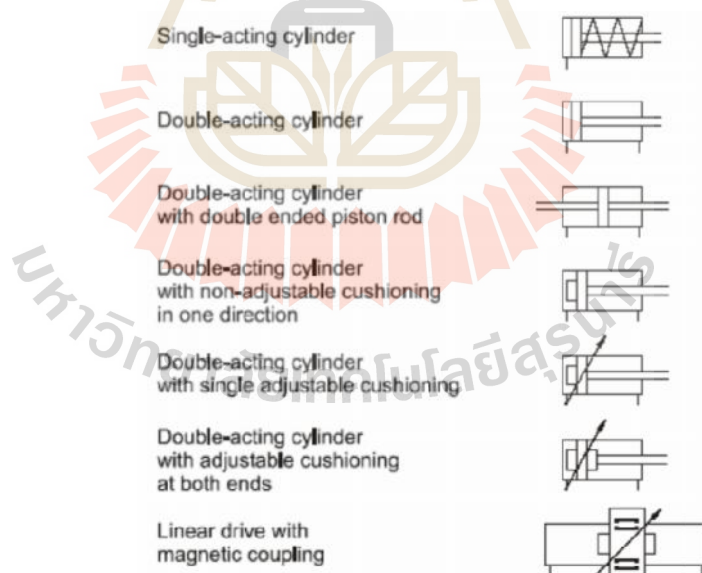
กระบอกสูบแบบสองก้านสูบแสดงดังรูปที่ 2.19 (ก) ซึ่งก็คือกระบอกสูบปกติที่มีก้านสูบยาวทะลุทั้งสองด้านสามารถเคลื่อนเข้าและออกสลับด้านกันเสมอ กระบอกสูบแบบช่วงชักหลายตำแหน่ง (Multi-Position cylinder) เป็นการนำกระบอกสูบสองตัวต่อกันดังรูปที่ 2.19 (ข) ทำให้เกิดการเคลื่อนที่ได้ 4 ตำแหน่ง



(ก) กระบอกสูบแบบสองก้าน



(ข) กระบอกสูบแบบหลายช่วงชัก



รูปที่ 2.19 กระบอกสูบชนิดพิเศษและสัญลักษณ์กระบอกสูบชนิดต่างๆ

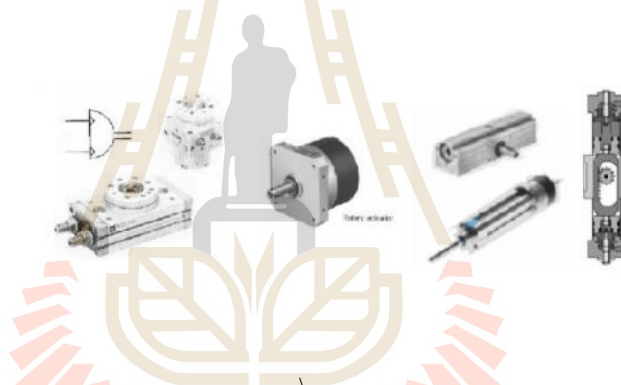


#### 2.5.1.4 กระบอกลูกสูบโรตารี Rotary Actuator

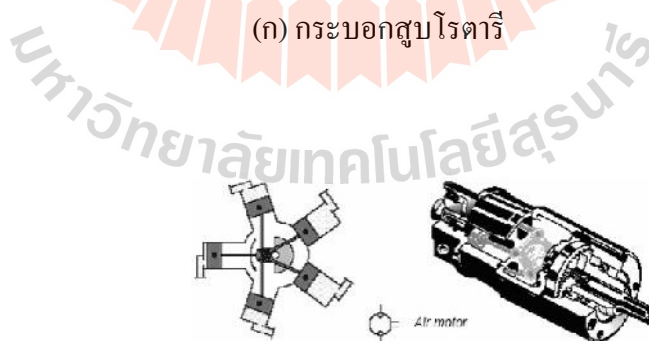
กระบอกลูกสูบชนิดนี้เมื่อจ่ายลมอัดเข้าภายในกระบอกลูกสูบ ก้านสูบจะเกิดการหมุนไป-มาได้ แต่ไม่สามารถหมุนรอบตัวมากกว่า 360 องศาได้ สามารถควบคุมการหมุนได้ทั้งสองทิศทางคือตามเข็มนาฬิกา และทวนเข็มนาฬิกา ในการเลือกใช้งานสามารถกำหนดมุมในการกวาด ด้วยการปรับตั้งสลักที่ฐานหมุน

#### 2.5.1.5 มอเตอร์ลม Motor

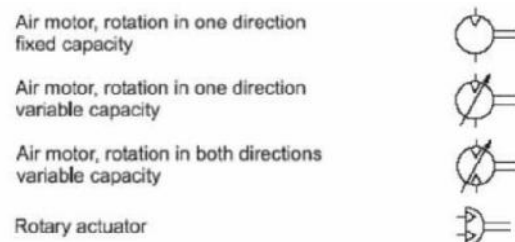
เป็นอุปกรณ์ที่มีแกนกลางหมุนได้ด้วยลมอัด นิยมใช้ในงานที่มีการกระแทกสูงเช่น ส่วนเจาะปูนซีเมนต์หรือในบริเวณที่ไม่ต้องการให้เกิดประกายไฟจากมอเตอร์ไฟฟ้า โดยข้อแตกต่างระหว่างมอเตอร์ลมและกระบอกลูกสูบโรตารีคือ กระบอกลูกสูบโรตารีไม่สามารถหมุนเกิน 360 องศาได้แต่มอเตอร์ลมสามารถหมุนรอบได้เหมือนมอเตอร์ไฟฟ้า



(ก) กระบอกลูกสูบโรตารี



(ข) มอเตอร์ลม

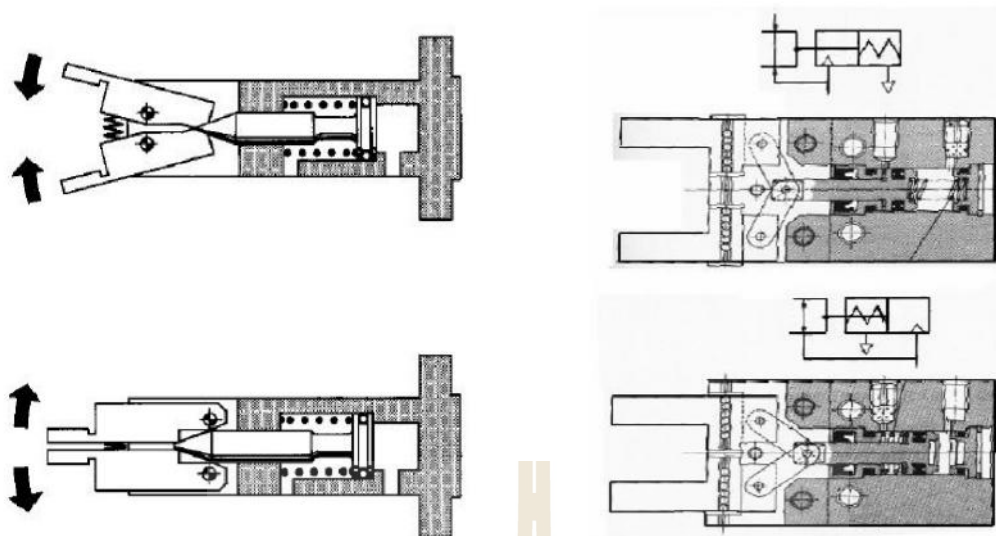


รูปที่ 2.20 กระบอกสูบโรตารีมอเตอร์และสัญลักษณ์

### 2.5.1.6 อุปกรณ์หยิบจับ Grippers

เป็นการอุปกรณ์ในการจับยึดชิ้นงานโดยใช้กลไกและกระบอกสูบทำงานร่วมกัน สามารถหยิบจับชิ้นงานรูปทรงต่างๆ ดังรูปที่ 2.21 และแสดงหลักการทำงานดังรูปที่ 2.22 ซึ่งนิยมใช้งานร่วมกับระบบอัตโนมัติและหุ่นยนต์อุตสาหกรรม แสดงดังรูปที่ 2.23



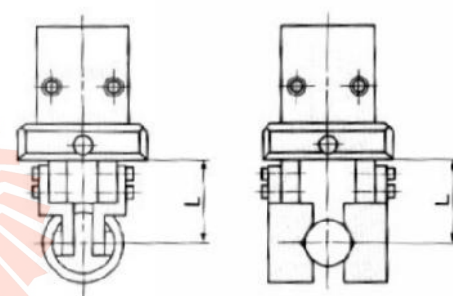


รูปที่ 2.21 ตัวอย่างอุปกรณ์หยิบจับจากยี่ห้อ Festo

a) ปลายจับของขนาดเล็กแบบหนีบ



b) ปลายจับของขนาดใหญ่แบบเลื่อน

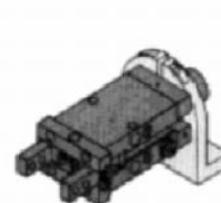
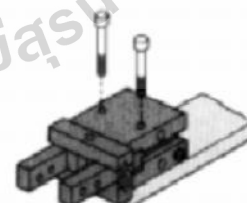
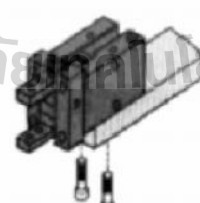
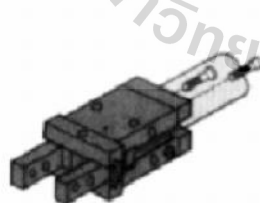


c) หยิบจับ 3 นิ้ว

d) ปลายจับหมุนได้

e) จับที่กลมกลวง

f) จับที่กลมตัน



Axial mounting

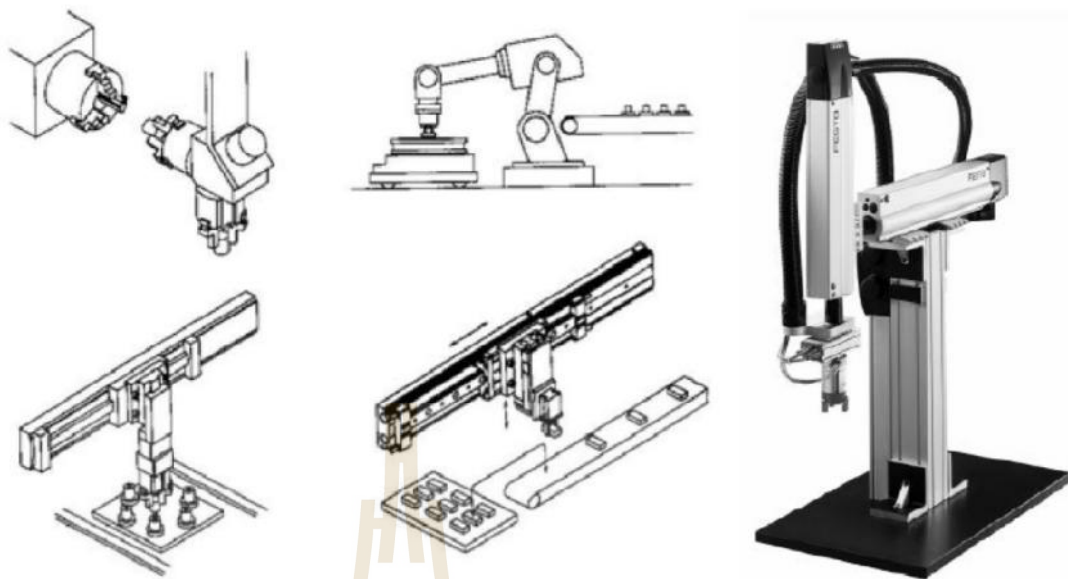
Vertical mounting

Lateral mounting

End boss

g) ลักษณะการจัดยึด Grippers

รูปที่ 2.22 หลักการทำงานภายในและการจับยึดชิ้นงานรูปทรงต่างๆ



รูปที่ 2.23 ตัวอย่างการประยุกต์อุปกรณ์หีบจับที่ใช้งานร่วมกับหุ่นยนต์อุตสาหกรรม

### 2.5.2 วาล์วควบคุมในระบบนิวเมติกส์



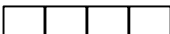
วาล์วควบคุมในระบบนิวเมติกส์พื้นฐานนั้นแบ่งได้ 5 ประเภท โดยในแต่ละประเภทแตกต่างกันตามลักษณะการใช้งาน ในบทนี้จะกล่าวเฉพาะวาล์วควบคุมทิศทาง (Directional Control) ที่ใช้สำหรับควบคุมการเคลื่อนที่ของกระบอกสูบ โดยวาล์วประเภทอื่นๆจะกล่าวถึงในบทถัดไป อาทิเช่น วาล์วชนิดลมไหลทางเดียวในการออกแบบสร้างวงจรควบคุมการทำงานแบบมีเงื่อนไขและวงจรควบคุมเพื่อความปลอดภัยของผู้ใช้งาน การใช้วาล์วควบคุมอัตราการไหลในการควบคุมความเร็วของก้านสูบ การใช้วาล์วแบบผสมที่ออกแบบสำหรับงานเฉพาะ เป็นต้น

- วาล์วควบคุมทิศทาง (Directional control valves)
- วาล์วชนิดลมไหลทางเดียว (Non-return valves)
- วาล์วควบคุมความดัน (Pressure control valves)
- วาล์วควบคุมอัตราการไหล (Flow control valve)
- วาล์วเปิด-ปิดและวาล์วผสม (Shut-of valve and Valve combination)

### 2.5.2.1 วาล์วควบคุมทิศทาง Directional control valves (D.C.V)

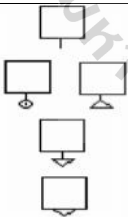
วาล์วควบคุมทิศทางมีหน้าที่ในการควบคุมทิศทางของลมอัดที่จ่ายให้กับอุปกรณ์นิวเมติกส์ โดยภายในประกอบมีลิ้นวาล์วที่เคลื่อนที่ได้ ตำแหน่งของลิ้นวาล์วที่เคลื่อนที่ได้นี้จะเรียกว่า “ตำแหน่งทำงาน (Position)” ใช้สัญลักษณ์รูปสี่เหลี่ยมแทนจำนวนตำแหน่งที่ลิ้นวาล์วที่สามารถเปลี่ยนได้ ตามข้อกำหนดการเรียกชื่อวาล์วควบคุมตามมาตรฐาน ISO 1219 โดยความหมายของสัญลักษณ์ตำแหน่งทำงาน (Position) จะดูได้จากจำนวนช่องสี่เหลี่ยมซึ่งจะหมายถึงจำนวนตำแหน่งทำงานที่ลิ้นวาล์วเปลี่ยนได้

ตารางที่ 2.3 สัญลักษณ์ของจำนวนตำแหน่งทำงานของวาล์ว

สัญลักษณ์	ความหมาย
	ลิ้นวาล์วเปลี่ยน 2 ตำแหน่งทำงาน
	ลิ้นวาล์วเปลี่ยน 3 ตำแหน่งทำงาน
	ลิ้นวาล์วเปลี่ยน 4 ตำแหน่งทำงาน









### 2.5.2.2 ความหมายของสัญลักษณ์ของช่องต่อท่อลม (Port)

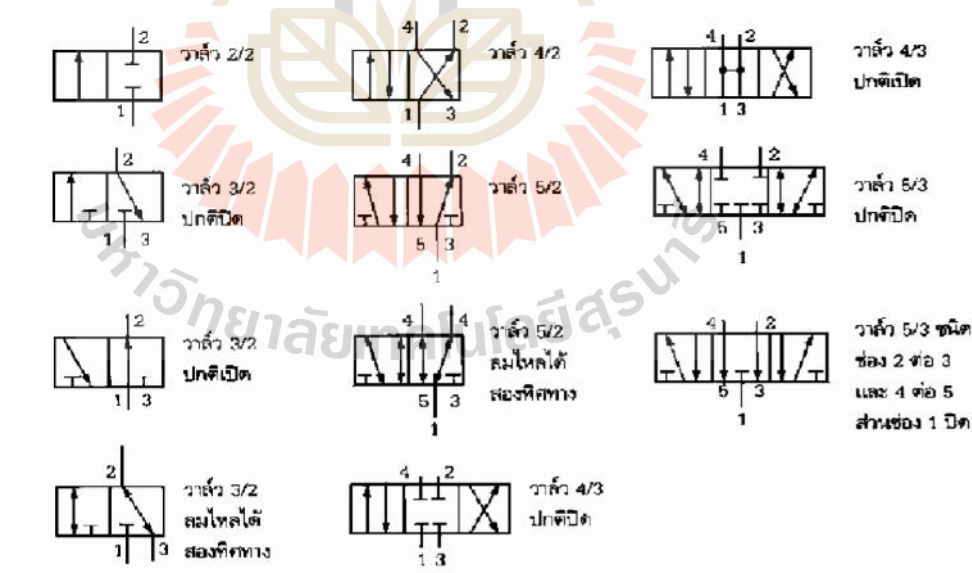
ตารางที่ 2.4 ความหมายของสัญลักษณ์ช่องต่อท่อลม

สัญลักษณ์	ความหมาย
	ช่องต่อสำหรับรับท่อลม
	ช่องต่อสำหรับจ่ายลมอัดให้วาล์ว
	ช่องต่อสำหรับระบายลมทิ้งแบบเปิด (มีช่องต่อท่อลม)
	ช่องระบายลมทิ้งแบบปิด (ไม่มีช่องต่อท่อลม)

### 2.5.2.3 สัญลักษณ์เส้นทางการไหลผ่าน

ตารางที่ 2.5 สัญลักษณ์เส้นทางการไหลผ่านภายในวาล์วที่แสดงด้วยลูกศรและช่องปิด

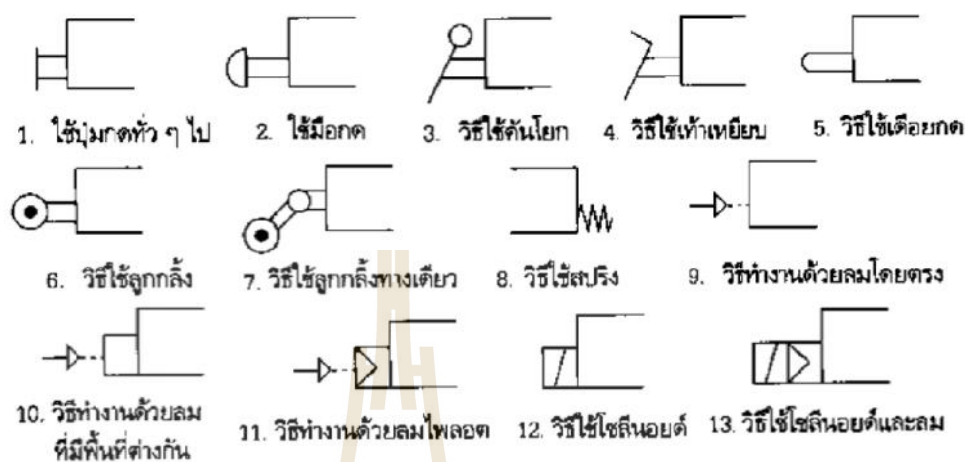
สัญลักษณ์	ความหมาย
	เส้นทางในรูปสี่เหลี่ยมหมายถึง เส้นทางที่ลมอัดไหลผ่านได้ตามทิศทางลูกศร
	เส้นทางเดินลมที่ถูกกั้นแสดงด้วยเส้นตรงตัดสั้นหัวท้าย
	เส้นทางเดินลมภายใน 2 เส้นทางที่ไม่ได้ตัดกัน และมีช่องต่อท่อลม 4 ช่อง
	เส้นทางเดินลมผ่านได้ทางเดียวและมีช่องต่อท่อลม 3 ช่อง
	เส้นทางเดินลมผ่านได้ทางเดียวและมีช่องต่อท่อลม 3 ช่อง
	เส้นทางเดินลมผ่านได้สองทางและมีช่องต่อท่อลม 5 ช่อง
	เส้นทางเดินลมผ่านได้สองทางและมีช่องต่อท่อลม 5 ช่อง
	เส้นทางเดินลมที่ต่อถึงกันภายในเขียนด้วยจุดตัดเต็มและมีช่องต่อท่อลม 4 ช่อง



รูปที่ 2.24 สัญลักษณ์และการเรียกชื่อวาล์วชนิดต่างๆ



รูปแบบการควบคุมวาล์ว Type of Control ในการควบคุมการทำงานของวาล์ว สามารถเลือกได้หลายรูปแบบ ดังแสดงในรูปที่ 2.26 และแสดงรูปแบบควบคุมแต่ละชนิดดังตารางที่ 2.5



รูปที่ 2.25 สัญลักษณ์ของรูปแบบที่ใช้ในการควบคุมวาล์ว

## 2.6 พื้นฐานระบบรับภาพและระบบการสอบเทียบภาพ

วิธีการสอบเทียบระบบรับภาพเชิงพื้นที่ (Spatial calibration) จะทำให้เราสามารถแปลงตำแหน่งโคออร์ดิเนตพิกเซล ให้เป็นโคออร์ดิเนตของโลกรจริงได้ ซึ่งจะสามารถแก้ไขความบิดเบี้ยว (Distortion) ที่เกิดจากระบบรับภาพ อันมีประโยชน์อย่างมากในการวัดละเอียด เพื่อให้ได้ค่าการวัดในหน่วยการวัดของโลกรจริง เช่น มิลลิเมตร เซนติเมตร และนิ้ว เป็นต้น นอกจากนี้ผลการสอบเทียบยังสามารถนำไปประยุกต์ในการสร้างภาพ 3 มิติในงานด้านวิชันหุ่นยนต์ (Robot Vision) ได้

### 2.6.1 ระบบรับภาพ

ก่อนการนำภาพไปประมวลผลนั้น สิ่งหนึ่งที่สำคัญคือ การออกแบบระบบรับภาพ (Image Acquisition System) ให้เหมาะสม เพื่อให้ได้ภาพที่มีคุณภาพที่ดีเพียงพอสำหรับการประมวลผล ระบบการรับภาพที่ดีจะทำให้การประมวลผลนั้นถูกต้อง และมีประสิทธิภาพสูง

#### 2.6.1.1 ตัวแปรที่สำคัญของระบบรับภาพ

เพื่อให้ได้ระบบรับภาพที่มีคุณภาพเราจำเป็นต้องมีความเข้าใจในตัวแปรที่สำคัญของระบบซึ่งประกอบด้วยค่าต่างๆดังนี้

##### - ความละเอียด

ในระบบรับภาพเราสามารถแบ่งความละเอียดได้เป็น 2 ชนิดคือ

- ความละเอียดพิกเซลซึ่งหมายถึงจำนวนของพิกเซลที่มีภาพ และมีกระบุเป็นจำนวนพิกเซลในแนวนอนคูณกับจำนวนพิกเซลในแนวตั้ง
- ความละเอียดหรือการแยกชัด หมายถึงระยะที่น้อยที่สุดที่ครอบคลุมลักษณะเฉพาะที่เล็กที่สุดที่สำคัญต่อการวัด หรือการตรวจจับสามารถเรียกสั้นๆ ว่า ฟีเจอร์ (Feature) โดยทั่วไปแล้วเราควรออกแบบให้อย่างน้อย 2 พิกเซลครอบคลุมฟีเจอร์ที่เล็กที่สุดที่เราต้องการ ดังสมการต่อไปนี้

$$N = \frac{L}{S} \times 2 \quad (2.7)$$

โดยที่ N คือจำนวนพิกเซล

L คือ ขนาดของวัตถุที่ยาวที่สุด

S คือ ขนาดของฟีเจอร์ที่เล็กที่สุด

ถ้าวัตถุมีขนาดเล็กกว่าขอบเขตของกล้องที่เก็บภาพได้ และถ้าความละเอียดของพิกเซลต่ำ ขนาดของภาพอาจใหญ่เกินความจำเป็น

#### - ขอบเขตของกล้องที่เก็บภาพได้

ขอบเขตที่กล้องเก็บภาพได้ (Field of View) หรือ FOV คือ

ขอบเขตของโลกจริงที่สามารถรับภาพได้ ดังรูปที่ 2.26

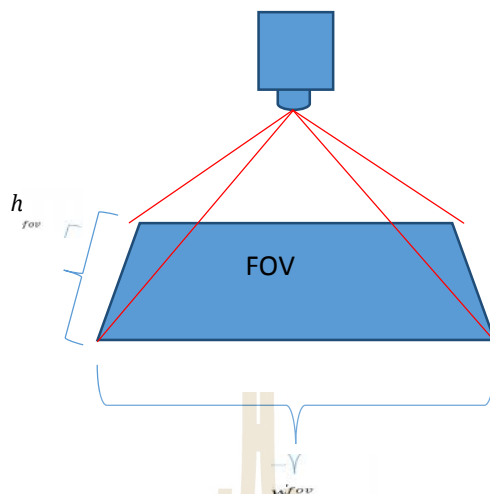
ความละเอียดพิกเซลในแนวนอนคือ

$$\frac{w_{fov}}{w} \times 2 \quad (2.8)$$

ความละเอียดพิกเซลแนวตั้งคือ

$$\frac{h_{fov}}{h} \times 2 \quad (2.9)$$





รูปที่ 2.26 ขอบเขตที่กล้องรับภาพได้ หรือ FOV

โดยที่  $w_{fov}$  คือขอบเขตของกล้องแนวนอน  $w$  ความยาวของฟิเจอร์แนวนอน  $h_{fov}$  คือขอบเขตของกล้องแนวตั้งและ  $h$  คือ ความยาวฟิเจอร์แนวตั้ง โดยมีมิติที่ให้ความละเอียดพิกเซลมากที่สุดจะเป็นมิติวิกฤตที่ใช้ในการออกแบบระบบรับภาพ

#### -ขนาดของเซนเซอร์และจำนวนพิกเซลของเซนเซอร์

ขนาดเซนเซอร์ของกล้อง และอัตราการขยายของเลนส์ เป็นตัวกำหนดขนาดของขอบเขตที่กล้องเก็บภาพได้ ซึ่งมีผลโดยตรงต่อค่าความละเอียดพิกเซลดังที่กล่าวมาแล้วในการพิจารณาเพื่อเลือกกล้องของระบบรับภาพ

#### -ระยะโฟกัสของเลนส์

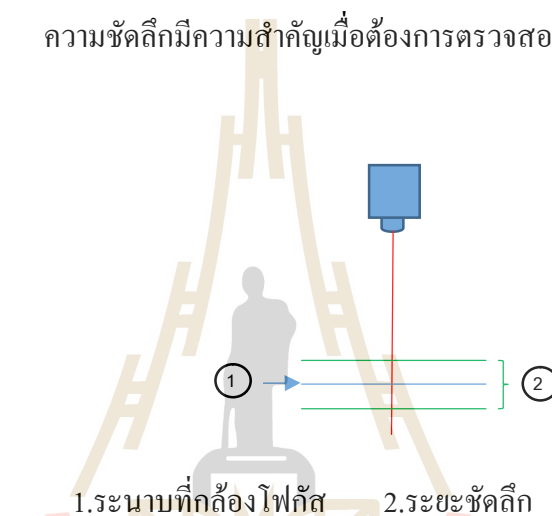
โดยปกติแล้วเราควรเลือกเลนส์ที่มีค่าโฟกัสยาวๆ เนื่องจากระยะชัดลึก (Depth of Field) ดังแสดงในรูปที่ 2.27 เป็นค่าพารามิเตอร์ของเลนส์ที่บ่งบอกถึงความสามารถในการให้ภาพที่ชัดของวัตถุ ในระยะที่ระยะต่างๆ ที่ไกล การใช้เลนส์ที่มีค่าระยะ โฟกัสสั้นจะส่งผลให้ภาพเกิดความผิดเพี้ยน (Distortion) มากขึ้น ซึ่งอาจทำให้เราต้องสอบเทียบภาพ หรือแก้ไขภาพก่อนการทำการวัด ซึ่งต้องเป็นวิธีที่ต้องใช้การคำนวณสูงขึ้น อย่างไรก็ตาม ถ้าเราไม่สามารถเปลี่ยนแปลงระยะห่างของระบบรับภาพได้ เราอาจจำเป็นต้องใช้การสอบเทียบเพื่อแก้ไขความบิดเบี้ยวของภาพ

### -ความคมชัด

ความคมชัด (Contrast) เป็นความแตกต่างระหว่างความเข้มแสงในบริเวณที่สว่างและมีดภายในภาพ โดยที่ระบบภาพที่ดีควรมีค่าความคมชัดสูง ตัวแปรหลักของอุปกรณ์รับภาพที่มีผลต่อค่าความคมชัดของภาพคือ กล้อง กับเลนส์ อย่างไรก็ตามเราสามารถให้แสงเพื่อเพิ่มคุณภาพของภาพได้

### -ระยะชัดลึก

ความชัดลึกมีความสำคัญเมื่อต้องการตรวจสอบวัตถุที่มีความสูงต่างกัน



รูปที่ 2.27 ระยะชัดลึกของเลนส์

### 2.6.2 ความบิดเบี้ยวจากระบบรับภาพ

ในระบบแมชชีนวิชันโดยทั่วไปเรามักพบความบิดเบี้ยวในภาพ ซึ่งเกิดจากระบบรับภาพความบิดเบี้ยวนี้ มีหลายรูปแบบ และเกิดจากหลายสาเหตุ โดยความบิดเบี้ยวที่สำคัญ และส่งผลมากต่อภาพมี 3 ประเภทคือ

**ความบิดเบี้ยวเพอร์สเปกทิฟ (Perspective distortion)** ความบิดเบี้ยวแบบเพอร์สเปกทิฟ เป็นความบิดเบี้ยวที่เกิดจากการตั้งระบบกล้อง แล้วแกนแสงไม่ตั้งฉากกับระนาบวัตถุ หรือระนาบวัตถุ ทำให้นขนาดของวัตถุมีขนาดต่างกัน ขึ้นอยู่กับระยะห่างของวัตถุกับกล้อง

**ความบิดเบี้ยวแนวรัศมี (Radial distortion)** ความบิดเบี้ยวแนวรัศมี เป็นความบิดเบี้ยวที่เกิดจากลักษณะความโค้งของเลนส์ ซึ่งจะส่งผลให้วัตถุที่เกิดในภาพมีความผิดเพี้ยน โดยที่รูปร่างของวัตถุ หรือระนาบมีรูปทรงแตกต่างจากความเป็นจริง

**ความบิดเบี้ยวแนวสัมผัส (Tangential distortion)** ความบิดเบี้ยวแนวเส้นสัมผัส เป็นความบิดเบี้ยวที่เกิดจากความไม่ขนานกัน ของระนาบเซนเซอร์กับระนาบของเลนส์ ซึ่งเกิดจากคุณภาพของระบบรับภาพที่ไม่ดี ส่งผลให้วัตถุในภาพมีตำแหน่งและรูปร่างต่างไปจากที่ควรจะเป็น

เพื่อให้ผลของการวัดมีคุณภาพสูง ผู้ใช้งานต้องแก้ไขข้อผิดพลาดดังกล่าว ด้วยการพยายามติดตั้งกล้องในระนาบให้ตั้งฉากกับวัตถุ และสอบเทียบระบบรับภาพ เพื่อแก้ไขหรือลดข้อผิดพลาดดังกล่าวมาข้างต้น ให้ได้มากที่สุดเท่าที่จะเป็นไปได้

### 2.6.3 การสอบเทียบระบบรับภาพเชิงพื้นที่

NI Vision ได้ให้การสอบเทียบระบบรับภาพเชิงพื้นที่ (Spatial Calibration) สำหรับสถานการณ์ที่แตกต่างกัน โดยแต่ละวิธีมีการใช้ระบบรับภาพที่แตกต่างกัน ขึ้นอยู่กับชนิดของการตรวจสอบและลักษณะของงานที่ตรวจสอบซึ่งมีระบบดังต่อไปนี้

- การสอบเทียบโดยใช้โมเดลระบบรับภาพอย่างง่าย ใช้ในการตรวจนับ หรือคัดกรองวัตถุที่มีขนาดต่าง ๆ กัน
- การสอบเทียบโดยใช้โมเดลความบิดเบี้ยว ใช้ในกรณีที่เลนส์มีความบิดเบี้ยวสูง
- การสอบเทียบโดยใช้โมเดลแบบเพอร์สเปกทีฟ ใช้ในกรณีที่แกนของแสงไม่ตั้งฉากกับระนาบวัตถุ
- การสอบเทียบโดยใช้โมเดลระบบรับภาพกล้องรูเข็ม ใช้ในการวัด และการตรวจสอบในลักษณะ 3 มิติ
- การสอบเทียบโดยใช้โมเดลไมโครเพลน ใช้ในการวัดหรืองานตรวจสอบที่ระนาบวัตถุเป็นผิวโค้ง

ข้อควรระวังหลังจากสอบเทียบระบบรับภาพแล้วคือ ต้องไม่กระทำการใดๆ ที่ทำให้เกิดความเปลี่ยนแปลงความสัมพันธ์ทางกายภาพในระบบรับภาพเช่นการปรับระยะโฟกัส การปรับระนาบวัตถุ การซูมหรือการปรับขนาดรูรับแสงของเลนส์

## 2.7 พื้นฐานของภาพดิจิทัล

### 2.7.1 พื้นฐานสามประการของภาพดิจิทัล

- ความละเอียดของภาพ (image resolution) คือค่าความละเอียดเชิงพื้นที่ (spatial resolution) ซึ่งเขียนได้เป็น  $m \times n$  โดยที่  $m$  คือจำนวนของพิกเซลในแนวนอน และ  $n$  คือจำนวนของพิกเซลในแนวตั้ง

- Bit depth คือ ค่าที่บอกถึงความเป็นไปได้ของเฉดสีในภาพ โดยที่ bit depth จะระบุเป็นจำนวนบิตในการเข้ารหัสความเข้มของ 1 พิกเซล หากค่าของ Bit depth ที่ค่าเท่ากับค่า  $k$  ค่าของจำนวนเฉดสีที่สามารถเป็นไปได้อาจมีค่าเท่ากับ  $2^k$  ดังนั้นหากค่า  $k$  มีค่าเท่ากับ 8 บิตหมายความว่าพิกเซลแต่ละพิกเซลจะสามารถแสดงค่าของสีได้  $2^8$  นั่นเท่ากับ 256 ค่าที่แตกต่างกัน โดยที่ค่าที่ได้จะอยู่ในช่วง 0- 255 ค่าและถ้าหากค่าของ  $k$  มีค่า Bit depth เท่ากับ 16 บิตหมายความว่าแต่ละพิกเซลจะแสดงค่าของสีได้แตกต่างกันทั้งหมด  $2^{16}$  นั่นเท่ากับ 65536 ค่า โดยค่าที่ได้จะอยู่ในช่วง 0-65535 หรือ -32768 ถึง 32767 ในโปรแกรม NI vision สามารถประมวลผลภาพ 8 บิต 10 บิต 12 บิต 14 บิต 16 บิต และเลขทศนิยม (Floating point) หรือภาพที่มีการเข้ารหัสสีได้ โดยที่รูปแบบของภาพจะขึ้นอยู่กับอุปกรณ์รับภาพ ประเภทของการวิเคราะห์ภาพ และประเภทของการประมวลผลภาพ ถ้าหากการประมวลผลภาพที่เราสนใจแค่ความเข้มของแสง เราใช้การเข้ารหัสภาพ 8 บิต ก็เพียงพอที่ได้ข้อมูลที่เหมาะสม แต่ถ้าหากต้องการการประมวลผลที่มีความละเอียดมากยิ่งขึ้น ก็อาจจำเป็นต้องมีการเข้ารหัสภาพที่มีความสามารถสูงขึ้นเช่น การเข้ารหัสภาพ 16 บิต ในส่วนที่เราสนใจโดยที่สีของวัตถุภายในภาพ ก็จำเป็นต้องใช้การเข้ารหัสภาพสี

- จำนวนของระนาบในภาพ คือ ค่าที่มีความสัมพันธ์กับ จำนวนของอะเรย์ที่ใช้ในการเก็บค่าของภาพ ภาพระดับเทา (Gray Scale Image) จะมีจำนวนของระนาบเพียงหนึ่งระนาบ ในขณะที่ภาพสี (Color Image) จะมีจำนวนของระนาบทั้งหมด 3 ระนาบ ซึ่งแต่ละระนาบจะขึ้นอยู่กับปริภูมิสี (Color Space) เช่น ในระบบ RGB คือ ระนาบสีแดง (Red) ระนาบสีเขียว (Green) และระนาบสีน้ำเงิน (Blue) ส่วนในระบบของ HSV จะหมายถึง ระนาบ Hue ระนาบ Saturation และระนาบ Luminance

### 2.7.2 ชนิดของภาพดิจิทัล

โดยทั่วไปแล้วฟังก์ชันในไลบรารี NI Vision สามารถจัดการกับภาพได้ 3 ชนิด คือ ภาพระดับเทา ภาพสี และ ภาพเชิงซ้อน อย่างไรก็ตามฟังก์ชันของ NI Vision ก็ไม่สามารถประมวลผลภาพบางชนิดได้ เช่น การใช้ตัวดำเนินการ AND กับภาพเชิงซ้อน ทั้งนี้ขึ้นอยู่กับลักษณะการเข้ารหัสภาพในแต่ละแบบซึ่งจำแนกได้ดังนี้

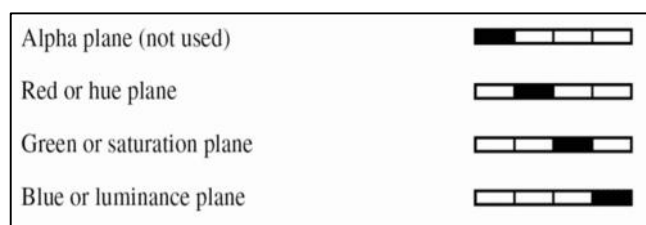
### 2.7.2.1 ภาพระดับสีเทา

ภาพที่เข้ารหัสแบบระดับสีเทานั้นจะประกอบด้วยพิกเซลที่อยู่ในระนาบ 1 ระนาบ โดยแต่ละพิกเซลอาจเข้ารหัสโดยใช้เลขจำนวนเต็มใด ๆ ได้ตามวิธีการต่อไปนี้

- ใช้ 8 บิต แบบไม่มีเครื่องหมาย (8-Bit Unsigned Integer) เป็นเลขจำนวนเต็มระหว่าง 0 ถึง 255 แทนค่าระดับความเข้มของสีเทา
- ใช้ 16 บิต แบบมีเครื่องหมาย (16-Bit Signed Integer) มีค่าระดับความเป็นสีเทาระหว่าง -32,768 ถึง 32,767
- ใช้เลขทศนิยมแบบความแม่นยำเดียว (Single-Precision Floating Point Number) เข้ารหัสโดยใช้ 4 ไบต์ หรือ 32 บิต

### 2.7.2.2 ภาพสี

การเข้ารหัสเพื่อให้ได้ภาพสีนั้น จะมีการเข้ารหัสระดับความเข้มของสี 3 สี คือ สีแดง สีเขียว และสีน้ำเงิน ในระบบ RGB หรือในอีกวิธีการหนึ่งจะเป็นการแสดงด้วยค่า Hue Saturation และ Luminance หรือภาพแบบ HSL โดยในการเข้ารหัสของภาพสีทั้งสองแบบ จะเป็นการเข้ารหัสที่ให้แต่ละพิกเซลมีค่าลำดับ 4 ค่า สำหรับภาพ RGB จะเก็บค่าโดยใช้เลข 8 บิต เพื่อเก็บค่าสี R-G-B ในแต่ละระนาบ ส่วน HSL ก็จะใช้เลข 8 บิตสำหรับค่า H-S-L ของแต่ละพิกเซลเช่นกันดังแสดงในรูปที่ 2.28 สำหรับภาพเข้ารหัสแบบ RGB-U64 จะเป็นการเก็บข้อมูลรหัสโดยใช้เลข 16 บิต สำหรับสี R-G-B ในแต่ละระนาบ อย่างไรก็ตามภาพสีที่ใช้อยู่โดยทั่วไปจะมีการเตรียมระนาบของสีไว้จำนวนทั้งสิ้น 4 ระนาบ ดังนั้นจะมีระนาบที่ไม่ได้ใช้งานในกรณีภาพสีบางแบบอยู่หนึ่งระนาบ และเนื่องจากการเตรียมระนาบไว้ 4 ระนาบ จึงมีการเรียกรูปสีเหล่านี้ว่าเป็นแบบ 4x8 บิต หรือการเข้ารหัสแบบ 32 บิต ส่วนภาพที่มีเฉฟฟินิชัน 16 บิต ก็จะเรียกว่าภาพ 4x16 บิตหรือการเข้ารหัสแบบ 64 บิต



รูปที่ 2.28 ระนาบที่บรรจุภาพสี

### 2.7.2.3 ภาพสี่เชิงซ้อน

ในการเข้ารหัสของภาพสี่เชิงซ้อน จะเป็นการบอกค่าความถี่ (Frequency) ของภาพระดับสีเทา ซึ่งการสร้างภาพสีจะสามารถทำได้โดยการเปลี่ยนโดเมนที่ภาพนั้นอยู่ เราจะสามารถสร้าง Complex Image ได้โดยใช้ Fast Fourier Transform (FFT) เข้ากับรูปที่ต้องการ และเมื่อเราเปลี่ยนค่าของภาพระดับสีเทา ให้ไปอยู่ในรูปของภาพสี่เชิงซ้อน แล้วเราจะสามารถใช้การคำนวณ หรือดำเนินการต่าง ๆ ในโดเมนความถี่เชิงซ้อน กับภาพที่เรากำลังวิเคราะห์ก็ได้ สำหรับในแต่ละพิกเซลของภาพเชิงซ้อน ภาพจะถูกเข้ารหัสด้วยตัวเลขทศนิยมความแม่นยำเดียว 2 จำนวน ซึ่งแต่ละจำนวนก็จะแทนค่าของส่วนจริง และอีกจำนวนหนึ่งจะแทนส่วนของจำนวนจินตภาพ ตามลำดับ เราสามารถทราบค่าที่สำคัญที่ประกอบด้วย 4 องค์ประกอบออกจากภาพเชิงซ้อน นั่นคือ ส่วนจริง ส่วนจินตภาพ ขนาด (Magnitude) และ เฟส (Phase)

### 2.7.3 ประเภทของไฟล์ภาพ (Image File)

ประเภทของไฟล์ภาพที่นิยมใช้ในการเก็บภาพสำหรับคอมพิวเตอร์นั้นมีหลายประเภท ซึ่งจะมีลักษณะการจัดเก็บขนาดของไฟล์ที่แตกต่างกัน แต่ในไฟล์รูปภาพจะมีส่วนประกอบสำคัญที่เหมือนกันหลายจุด ในส่วนแรกไฟล์จะประกอบด้วย หัวไฟล์ (Header) หรือหัวเรื่องซึ่งบอกลักษณะเฉพาะของไฟล์นั้น จากนั้นจะตามด้วยค่าพิกเซล ซึ่งรายละเอียดการกำหนดค่าต่างๆ จะขึ้นอยู่กับประเภท หรือรูปแบบของไฟล์ (File Format) แต่โดยทั่วไป หัวเรื่องของไฟล์จะบรรจุข้อมูลเกี่ยวกับไฟล์ เช่น จำนวนพิกเซลในแต่ละแถวและคอลัมน์ ค่าเดฟนิชันของพิกเซล และค่าอื่น ๆ นอกเหนือจากนั้นไฟล์รูปอาจจะบรรจุข้อมูลที่เกี่ยวข้องกับการปรับวัด (Calibration) รูปแบบการจับคู่ (Pattern Matching Templates) และ Overlay ซึ่งนิยามของข้อมูลเหล่านี้จะกล่าวในภายหลัง ส่วนรูปแบบไฟล์ดิจิทัลที่นิยมใช้กันมีดังต่อไปนี้

- Bitmap (BMP)
- Tagged Image File Format (TIFF)
- Portable Network Graphic (PNG) โดยไฟล์นี้จะให้ข้อมูลที่สำคัญหลาย

แบบเช่น การย่อ – ขยายรูปภาพ Calibration Pattern Matching และ Overlay

- Joint Photographic Experts Group Format (JPEG)
- National Instrument Internal Image File Format (AIPD) ใช้ในการเก็บ

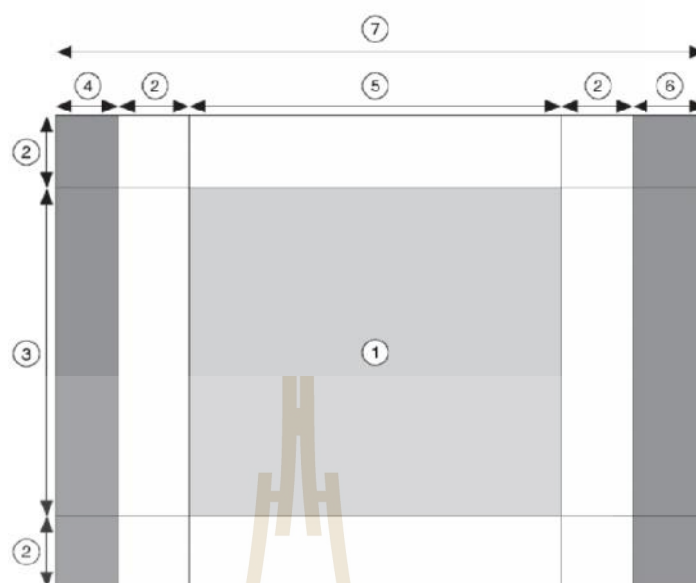
ไฟล์แบบเลขทศนิยม ภาพสี่แบบเชิงซ้อน และภาพสี่แบบ HSL

สำหรับโปรแกรมที่ใช้จะเป็นโปรแกรมที่สามารถรับรูปแบบไฟล์ได้หลายแบบ ประกอบด้วย รูปแบบไฟล์มาตรฐาน แบบระดับสีเทา 8 บิต ส่วนไฟล์ภาพสีแบบ RGB จะมีรูปแบบได้หลายแบบ คือ BMP TIFF PNG JPEG และ AIPD ส่วนมาตรฐานของระดับสีเทา 16 บิตระบบสี RGB แบบ 64 บิตและภาพเชิงซ้อน จะใช้ไฟล์ในรูปแบบของ PNG และ AIPD

#### 2.7.4 รูปแบบการเก็บภาพในหน่วยความจำของ NI Vision

เมื่อมีการนำภาพเข้ามาเพื่อทำการประมวลผลและวิเคราะห์ผล โปรแกรมจะปรับเปลี่ยนค่าบางส่วนของรูปนั้น เพื่อให้เหมาะกับการทำงานของโปรแกรม เมื่อมีการนำรูปภาพเข้ามาใช้โดยโปรแกรม NI Vision จะมีการปรับรายละเอียดให้เหมาะสมกับการทำงาน ซึ่งรายละเอียดภายในของโปรแกรม เมื่อพิจารณารูปภาพจะเป็นไปในลักษณะดังแสดงในรูปที่ 2.29 ซึ่งแสดงให้เห็นว่า NI Vision มีกระบวนการจัดการรูปภาพในหน่วยความจำอย่างไรบ้าง นอกเหนือจาก Image Pixel แล้ว รูปภาพยังจะประกอบด้วยแถวบนและแถวตั้ง ที่เพิ่มเติมขึ้นมาอีก เรียกว่า ขอบภาพ (Image Border) และยังมีค่าการจัดชิดขอบซ้าย (Left Alignments) และค่าการจัดชิดขอบขวา (Right Alignments) สาเหตุที่ต้องมีการจัดทำองค์ประกอบภายในเช่นนี้ก็เพราะว่า ในฟังก์ชันบางแบบที่ใช้ในการประมวลผลภาพนั้น อาจจะต้องใช้ขอบของภาพในการประมวลผล และสำหรับในส่วนของพื้นที่จัดชิดขอบ (Alignment Regions) มีไว้เพื่อให้มั่นใจว่า พิกเซลแรกของจำนวน 32 ไบต์นั้น จะอยู่ในตำแหน่งหน่วยความจำที่ถูกต้อง โดยขนาดของการจัดขอบ (Alignment Blocks) จะขึ้นกับความกว้างของรูป และขนาดของขอบ การจัดเรียงภาพจะทำให้การประมวลผลภาพมีประสิทธิภาพ และความเร็วสูงสามารถเพิ่มขึ้นได้สูงสุดถึง 30% ความกว้างของเส้น (Line Width) หมายถึงจำนวนพิกเซลในเส้นตามแนวระดับในภาพ ซึ่งเป็นผลรวมของผลความละเอียดในแนวระดับ ขอบของภาพ และการจัดเรียงซ้าย-ขวา ค่าความละเอียดในแนวระดับ และความกว้างของเส้น อาจมีค่าเท่ากัน ถ้าหากว่าค่าความละเอียดในแนวระดับเป็นจำนวนเท่าของตัวเลข 32 ไบต์และขนาดของขอบมีค่าเท่ากับ 0





1. ข้อมูลภาพ
2. บริเวณขอบภาพ
3. ความละเอียดของภาพในแนวดิ่ง
4. ขนาดการทำให้เป็นแนวเดียวกันด้านซ้าย
5. ความละเอียดของภาพในแนวนอน
6. ขนาดการทำให้เป็นแนวเดียวกันด้านขวา
7. ขนาดแนวเส้นในหน่วยความจำ

รูปที่ 2.29 รูปแบบการเก็บภาพในหน่วยความจำของ NI Vision

## 2.8 พื้นฐานการประมวลผลภาพ

### 2.8.1 ขอบภาพ

ในการประมวลผลภาพส่วนใหญ่แล้วเมื่อมีการประมวลผลค่าที่พิกเซลใดๆ มักจะต้องใช้ค่าของพิกเซลที่อยู่รอบๆ ข้าง มาใช้ในการประมวลผลด้วย เช่น การดูว่าค่าความเข้มของภาพนั้นเพิ่มขึ้นหรือลดลง หรือการหาค่าเฉลี่ยของภาพบนพื้นที่ๆ ต้องการเป็นต้น พิกเซลที่อยู่รอบข้าง (Neighbors) จะเป็นพิกเซลที่มีผลกระทบต่อการคำนวณของค่าพิกเซลอื่นๆ ต่อไป อย่างไรก็ตาม พิกเซลที่อยู่ตามขอบ จะมีพิกเซลที่อยู่รอบข้างน้อยกว่าพิกเซลที่อยู่ห่างขอบไปหลายๆ เช่น พิกเซลที่อยู่ขอบซ้ายสุด จะไม่มีพิกเซลรอบข้างด้านซ้ายเลย ดังนั้นค่าที่ใช้ในการคำนวณของฟังก์ชันนั้นจึง



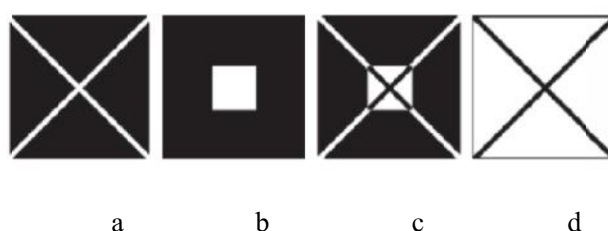
ไม่ครบถ้วน ทำให้โปรแกรมที่ใช้สำหรับการประมวลผลภาพทำการสร้างขอบของภาพเพิ่มเติมขึ้นมาโดยอัตโนมัติ ทั้งนี้เมื่อการประมวลผลภาพจึงต้องการค่าพิกเซลที่ใกล้เคียงพิกเซลที่กำลังประมวลผลอยู่ เพื่อจะได้มีค่าส่งไปคำนวณอย่างครบถ้วน ค่าขอบภาพจะกำหนดด้วยจำนวนพิกเซลที่ใช้เป็นขอบ และค่าของพิกเซลที่ขอบเหล่านั้น ขนาดของขอบควรจะต้องมีขนาดให้เหมาะสมกับฟังก์ชันในโปรแกรมที่จะใช้ในการคำนวณ นั่นคือต้องกำหนดให้มีขนาดจำนวนพิกเซลมากพอที่ฟังก์ชันที่เรียกใช้ จะสามารถนำพิกเซลที่อยู่ใกล้เคียงไปใช้ได้อย่างครบถ้วน ซึ่งขนาดของพิกเซลใกล้เคียงนี้จะบอกเป็นค่าลำดับในสองมิติ (2D Array) เช่น ถ้าฟังก์ชันที่เรียกใช้จำเป็นต้องใช้พิกเซลที่อยู่ใกล้กันจำนวน 8 พิกเซล ในการคำนวณ ขนาดของพิกเซลใกล้เคียงจะบอกเป็นเมตริกซ์ขนาด 3x3 ซึ่งหมายถึงจะมีพิกเซลในสามแถวตั้งและสามแถวนอน โดยพิกเซลที่อยู่ตรงกลางเป็นพิกเซลที่กำลังพิจารณา และพิกเซลที่เหลือเป็นพิกเซลใกล้เคียง ดังนั้นเราจึงควรกำหนดขนาดของขอบมากกว่า หรือเท่ากับครึ่งหนึ่งของจำนวนแถวตั้งหรือแถวนอน ในเมตริกซ์สองมิตินั้น ดังนั้นจากตัวอย่างที่ได้กล่าวมา ขนาดของขอบอย่างน้อยต้องเป็น 1 แต่ถ้าฟังก์ชันที่ใช้ต้องใช้พื้นที่เมตริกซ์ขนาด 5x5 ก็จะต้องมีความกว้างของขอบไม่น้อยกว่า 2 สำหรับโปรแกรมที่ใช้ในที่นี้คือ NI Vision จะมีการสร้างขอบให้อัตโนมัติ โดยขนาดของขอบจะสร้างให้มีขนาด 3 พิกเซล ดังนั้นจึงรองรับพื้นที่การวิเคราะห์ได้ในขนาด 7x7 โดยไม่ต้องมีการใช้ค่าตั้งใดเพิ่มเติม อย่างไรก็ตามในกรณีที่ผู้ใช้งานมีความต้องการขนาดของขอบมากกว่านี้ ผู้ใช้ก็สามารถที่จะปรับเปลี่ยนได้ด้วยตนเอง โดยโปรแกรม NI Vision มีวิธีการอยู่ 3 วิธีที่จะใช้ในการกำหนดค่าของพิกเซลที่ขอบที่โปรแกรมสร้างขึ้น ดังแสดงในรูปที่ 2.30 โดยการกำหนดค่าเริ่มต้นค่าของพิกเซลที่ขอบ ทั้งหมดจะมีค่าเท่ากับ 0 ดังแสดงในรูปที่ 2.30b อย่างไรก็ตามผู้ใช้สามารถที่จะนำค่าที่อยู่ในตำแหน่งที่ติดกับขอบ ไปใช้เป็นค่าที่ขอบได้เช่นกัน ดังแสดงในรูปที่ 2.30c หรือไม่เช่นนั้น อาจจะใช้วิธีการสะท้อน (Mirror) ค่าที่อยู่ติดกับแนวขอบให้ไปเป็นค่าที่ขอบก็ได้ ดังแสดงในรูปที่ 2.30d



กรอบ ให้มีค่าเป็นศูนย์ทั้งหมด อย่างไรก็ตามค่าที่กำหนดในส่วนพิกเซลที่ขอบนั้น มีไว้เพื่อใช้ในการคำนวณเท่านั้น ค่าเหล่านี้จะไม่มีผลแสดงผล หรือจัดเก็บในระบบไฟล์ของภาพเลย

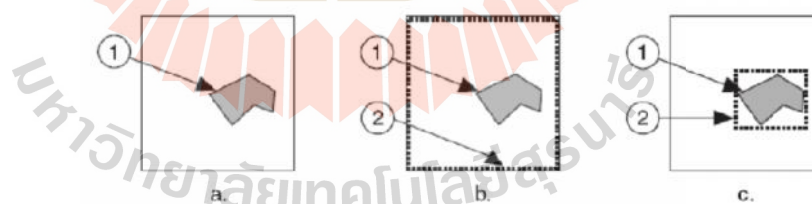
### 2.8.2 หน้ากากของภาพ (Image Mask)

หน้ากากของภาพจะเป็นส่วนของภาพเสริมที่จะนำไปประมวลผล ถ้าหากว่าฟังก์ชันที่ต้องการเรียกใช้นั้น ต้องการกำหนดค่าของหน้ากากของภาพ การประมวลผลหรือวิเคราะห์ผลจะกระทำในภาพต้นฉบับ โดยดึงข้อมูลการวิเคราะห์มาจากหน้ากากของภาพ หน้ากากของภาพจะเป็นภาพแบบไบนารีขนาด 8 บิต อาจมีขนาดเล็กกว่าหรือเท่ากับภาพที่ต้องการนำมาตรวจสอบก็ได้ พิกเซลที่อยู่บนหน้ากากของภาพจะทำการค้นหาว่าพิกเซลในตำแหน่งเดียวกันในภาพนั้น ต้องการที่จะได้รับการประมวลผลหรือไม่ ถ้าค่าของพิกเซลในหน้ากากของภาพมีค่าไม่เท่ากับศูนย์ พิกเซลที่ตำแหน่งเดียวกันที่อยู่ในภาพตรวจสอบจะได้รับการประมวลผล และแสดงค่าใหม่ที่ได้จากการประเมินผลในพิกเซลนั้น แต่ถ้าค่าของพิกเซลในหน้ากากของภาพมีค่าเท่ากับศูนย์ พิกเซลที่ตำแหน่งเดียวกันที่อยู่ในภาพตรวจสอบ จะไม่ต้องรับการประมวลผล ซึ่งการใช้งานหน้ากากของภาพก็เพื่อจะได้มีเป้าหมายในการพิจารณาภาพให้ชัดเจน ว่าต้องการพิจารณาที่ส่วนใดในภาพทั้งหมด บางหลักการของการจัดหน้ากากของภาพ ก็เพื่อทำให้การประมวลผลเกิดขึ้นในส่วนที่ผู้สนใจเท่านั้น จึงทำให้การประมวลผลกระทำได้รวดเร็วขึ้น การใช้หน้ากากของภาพจะกระทบกับผลที่ได้รับจากการประมวลผลของฟังก์ชันอย่างไร ดังแสดงในรูปที่ 2.31 สมมุติว่าฟังก์ชันที่ใช้เป็นฟังก์ชันที่ใช้ในการผกผันค่าของพิกเซล และมีภาพที่ต้องการจะตรวจสอบดังแสดงในรูปที่ 2.31a มีหน้ากากของภาพซึ่งมีขนาดเท่ากับภาพที่จะทำการตรวจสอบดังแสดงในรูปที่ 2.31b หากพิกเซลที่มีค่าเท่ากับศูนย์จะแสดงเป็นสีดำและพิกเซลที่มีค่าอื่นแสดงสีขาว จะได้ผลลัพธ์กรณีมีการกำหนดหน้ากากของภาพดังแสดงในรูปที่ 2.31c และจะแสดงผลลัพธ์ที่ได้ในกรณีที่ไม่มีกำหนดส่วนแบ่งของรูปดังแสดงในรูปที่ 2.31d



รูปที่ 2.31 หลักการทำงานของการจัดส่วนแบ่งภาพ

จากรูปที่ 2.31 นี้วิธีการที่จะกำหนดการใช้งานของฟังก์ชันจะสามารถวางกรอบการทำงานของโปรแกรมให้ดำเนินการทำงานหรือประมวลผลเฉพาะในส่วนที่ต้องการได้ วิธีการนี้จะเป็นการประหยัดหน่วยความจำของเครื่องให้มีขนาดที่ใช้ในการประมวลผลน้อยลงมาก เพราะจะเป็นการประมวลผลเฉพาะในส่วนของพื้นที่ซึ่งมีความสำคัญ และเป็นส่วนที่กำหนดให้ทำการประมวลผลเท่านั้น พื้นที่ในส่วนนี้อาจเรียกว่าเป็นพื้นที่ๆ เราสนใจหรือ Region of Interest (ROI) ภายในภาพ ดังนั้น ความแตกต่างของ ROI กับ Image Marks คือ ROI จะเป็นส่วนในภาพที่ต้องการทำการตรวจสอบ แต่ Image Marks จะเป็นภาพอีกภาพหนึ่งที่จะกำหนดว่าต้องการจะให้โปรแกรมตรวจสอบ ตรวจสอบพิกเซลใดบ้างใน ROI นอกเหนือจากนั้นโปรแกรม NI Vision ยังได้มีการกำหนดค่าเยื้อง (Offset) ให้กับหน้ากากของภาพ ค่าเยื้องนี้หมายถึงตำแหน่งตามระบบพิกัดในภาพของต้นแบบที่ต้องการตรวจสอบ ซึ่งจะวางระบบแกนของหน้ากากของภาพ และวิธีต่างๆ ที่มีการกำหนดหน้ากากของภาพดังแสดงในรูปที่ 2.32 มีการกำหนด ROI (Region of Interesting) ที่ต้องการที่จะสร้างเป็นหน้ากากของภาพขึ้นมาดังแสดงในรูปที่ 2.32a โดยหน้ากากของภาพมีขนาดเท่ากับภาพที่จะตรวจสอบดังแสดงในรูปที่ 2.32b ในกรณีนี้ค่าเยื้องจะกำหนดให้เป็น (0,0) ในอีกกรณีหนึ่ง หน้ากากของภาพนี้สามารถที่จะจัดให้เป็นเพียงกรอบสี่เหลี่ยมที่พอที่จะล้อมรอบรูปให้เป็นพื้นที่ ROI ดังแสดงในรูปที่ 2.32c ซึ่งการกำหนดค่าเยื้องจะกำหนดโดยตำแหน่งของหน้ากากของภาพ เทียบต่อภาพอ้างอิง นอกจากนี้ยังสามารถที่จะกำหนดค่าเยื้องให้กับหน้ากากของภาพในพื้นที่ต่างๆ ในภาพที่จะทำการตรวจสอบได้

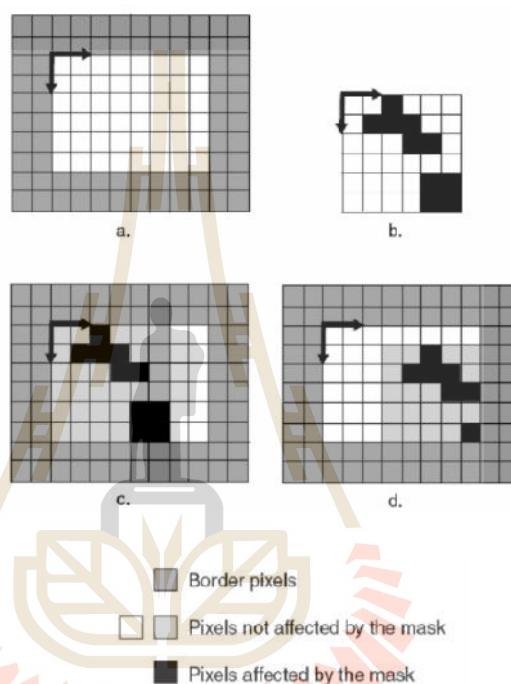


1) Region of Interest      2) Image Mask

รูปที่ 2.32 การกำหนดค่าเยื้องให้กับหน้ากากของภาพ

การใช้หน้ากากของภาพที่กำหนดค่าเยื้องที่แตกต่างกันสองแบบดังแสดงในรูปที่ 2.33 โดยแบบแรกเป็นการกำหนดตำแหน่งแกนอ้างอิง ที่อยู่ที่มุมบนด้านซ้ายของภาพที่ต้องการจะตรวจสอบดังแสดงในรูปที่ 2.12a และทำการกำหนดหน้ากากของภาพ และตำแหน่งของระบบพิกัดไว้ที่มุมบนซ้ายของภาพเช่นกันดังแสดงในรูปที่ 2.12b เมื่อวางหน้ากากของภาพอาจจะวางให้จุดกำเนิดของทั้งสอง

ระบบแกนวางทับกัน ซึ่งทำให้ค่าเชื่อมโยงเท่ากับ (0,0) ดังแสดงในรูปที่ 2.12c หรืออาจวางเยื้องไปที่ตำแหน่งอื่น ตัวอย่างเช่น การกำหนดค่าเชื่อมโยงให้กับหน้าภาพของภาพเท่ากับ (3,1) ดังแสดงในรูปที่ 2.12d ซึ่งจะทำให้พิกเซลที่จะได้รับการประมวลผลทั้งในสองกรณีนี้แตกต่างกัน ส่งผลให้ได้ผลลัพธ์ที่แตกต่างกันด้วย



รูปที่ 2.33 ผลกระทบของการกำหนดค่าเชื่อมโยงให้กับหน้าภาพของภาพ

## 2.9 ปัญหาการเดินทางของพนักงานขาย (Travelling salesman problem : TSP)

ปัญหาการเดินทางของพนักงานขายเป็นหนึ่งในปัญหาที่ได้รับความสนใจจากนักวิจัยอย่างต่อเนื่องและยาวนาน ในการพัฒนาวิธีในการหาคำตอบให้ดีและเร็ว โดยปัญหา TSP นี้เป็นปัญหาที่ทำการตัดสินใจหาเส้นทางการเดินทาง เมื่อมีเมืองหรือสถานที่ๆ ต้องเดินทางเป็นจำนวน  $N$  เมือง โดยการเดินทางจะต้องเดินทางไปทุกเมืองและกลับมาที่เมืองเริ่มต้น การเริ่มต้นพัฒนาปัญหา TSP เกิดขึ้นในช่วง ค.ศ.1800 โดยนักคณิตศาสตร์ชาวไอร์แลนด์ ชื่อ W. R. Hamilton และชาวอังกฤษชื่อ Thomas Kirkman ได้สร้างเกมที่มีชื่อว่า Icosian ซึ่งเกมนี้ผู้เล่นจะต้องหาวงกลมฮามิลโทเนียน (Hamiltonian Cycle) จากนั้นช่วง ค.ศ.1930 Karl Menger ชาวเวียนนาได้พัฒนาวิธีการเดินทางจากเมืองที่ใกล้ที่สุด (Nearest Neighbor Heuristics) ช่วงปี ค.ศ.1950-1960 ปัญหา TSP ได้รับความสนใจ

เป็นอย่างมาก ทำให้ได้เริ่มมีการพัฒนาตัวแบบทางคณิตศาสตร์ และวิธีแม่นยำตรงในการแก้ปัญหา ต่อมาในปี คศ. 1972 Richard M. K ได้พิสูจน์ว่าเกมสัว์กลมฮามิโทเนียนเป็นปัญหาประเภท NP-Complete ซึ่งทำให้ปัญหา TSP เป็นปัญหายากพวกเดียวกันนั่นคือ NP-Hard แสดงให้เห็นว่า TSP เป็นปัญหาที่ยากและยังไม่มีวิธีการแก้ปัญหาที่ใช้เวลาแบบโพลิโนเมียลในการแก้ปัญหาได้ ดังนั้น จึงมีนักวิจัยหาวิธีในการแก้ปัญหาแบบวิธีการ (Algorithm) ที่ได้คำตอบดีที่สุดและได้เวลาน้อยกว่า  $O(n!)$  เช่นงานวิจัยของ Michael Held, Richard M. Karp and Richard Bellman ได้นำเสนอการแก้ปัญหาโดยใช้ Dynamic Programming ซึ่งพิสูจน์แล้วว่าสามารถลดเวลาจาก  $O(n!)$  ให้เหลือเพียง  $O(n^2 \cdot 2^{(n-1)})$  ปัจจุบันได้มีการหาวิธีในการแก้ปัญหา TSP หลากหลายวิธีด้วยกันเช่น Branch and Bound, Genetic Algorithm, Tabu Search, Ant Colony Optimization, 2-Opt เป็นต้น แต่ในงานวิจัยนี้เราเลือกใช้ Branch and Bound ซึ่งมีข้อดีคือมีความรวดเร็วในการประมวลผล ร่วมกับวิธี 2-Opt ในการจัดการเส้นทาง ซึ่งจะนำเสนอในหัวข้อต่อไป

### 2.9.1 หลักการของ Branch and Bound

หลักการของ Branch and Bound คือ การสร้างเส้นทางเริ่มต้นสำหรับการเปรียบเทียบมาหนึ่งเส้นทางหลังจากนั้นพยายามหาเส้นทางใหม่ที่ดีกว่าหากได้เส้นทางใหม่ที่ดีกว่าให้นำเส้นทางนั้นมาเป็นเส้นทางหลักเพื่อเปรียบเทียบกับเส้นทางอื่นๆต่อไป และหากเส้นทางใหม่ที่เกิดขึ้นแม้เพียงบางส่วนที่ยาวกว่าให้ทำการยกเลิกเส้นทางนั้นและทำเช่นนี้เรื่อยๆจนกระทั่งได้เส้นทางที่ดีที่สุด

ตารางที่ 2.6 ระยะทางระหว่างจุดของการเดินทาง

	A	B	C	D
A	0	20	30	50
B	20	0	15	20
C	30	15	0	10
D	50	20	10	0

ตัวอย่างจากตารางข้างบนมีจุดทั้งหมด 4 จุด เริ่มด้วยการหาระยะทางหลักก่อนโดยให้จุดเริ่มต้นคือ A หลังจากนั้นคำนวณระยะทางจาก A ไปยัง B, C และ D ซึ่งจะได้ระยะทางดังนี้

$$A B = 20$$

$$A C = 30$$



$$A D = 50$$

ดังนั้นจึงเลือกเส้นทาง A B C ซึ่งสั้นที่สุดจากนั้นคำนวณระยะทางที่สั้นที่สุดจาก B ไปยังตำแหน่งอื่นๆ ซึ่งจะได้ระยะทางดังนี้

$$A B C = 20 + 15 = 35$$

$$A B D = 20 + 20 = 40$$

ดังนั้นจึงเลือกเส้นทาง A B C ซึ่งสั้นที่สุดจากนั้นคำนวณระยะทางที่คำนวณได้เปรียบเทียบกับเส้นทางอื่นๆที่เคยคำนวณไว้ คือ A C กับ A D จะเห็นว่าเส้นทาง A C สั้นกว่าจากนั้นคำนวณเส้นทางจากตำแหน่ง C ไปยังตำแหน่งอื่นๆ

$$A C B = 20 + 10 + 15 = 45$$

$$A C D = 20 + 10 + 10 = 40$$

ดังนั้นจะได้เส้นทาง A B C คือเส้นทางที่สั้นที่สุดหลังจากนั้นคำนวณระยะทางที่สั้นที่สุดจาก C ไปยังตำแหน่งอื่นๆซึ่งได้ระยะทางดังนี้

$$A B C D = 20 + 15 + 10 = 45$$

ดังนั้นจึงเลือกเส้นทาง A B C D หลังจากนั้นคำนวณระยะทางที่คำนวณได้เปรียบเทียบกับเส้นทางอื่นๆที่คำนวณเอาไว้คือ A D, A B D, A C B, A C D จะเห็นว่า A C D สั้นกว่าจากนั้นคำนวณเส้นทางจาก D ไปยังตำแหน่งอื่นๆ

$$A C D B = 20 + 10 + 10 + 20 = 60$$

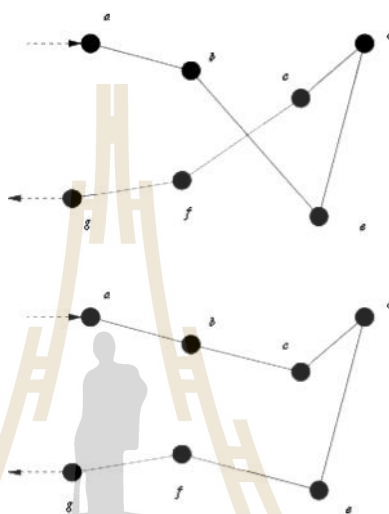
ดังนั้นจะได้เส้นทาง A B C D คือเส้นทางที่สั้นที่สุดคือ 45 เมื่อมีการเดินทางไปครบทุกตำแหน่ง อัลกอริทึมของ Branch and Bound คือ

1. หาระยะทางที่เป็นไปได้ทั้งหมดจากตำแหน่งเริ่มต้นไปยังตำแหน่งต่างๆเลือกตำแหน่งจากระยะทางที่สั้นที่สุดเพื่อใช้ในการสร้างเส้นทางที่สั้นที่สุด
2. หาระยะทางที่เป็นไปได้ทั้งหมดจากตำแหน่งที่เลือกและเลือกตำแหน่งจากระยะทางที่สั้นที่สุดเพื่อสร้างเส้นทางที่สั้นที่สุด
3. เปรียบเทียบระยะทางจากข้อที่ 2 กับระยะทางอื่นๆที่คำนวณมาแล้วจากข้อที่ 1 และ 2
4. หากมีเส้นทางที่สั้นกว่าให้ทำการสร้างเส้นทางใหม่เพื่อเปรียบเทียบและเลือกเส้นทางที่สั้นที่สุดจากนั้นทำซ้ำข้อที่ 2-5 จนกว่าจะครบทุกตำแหน่ง
5. จากข้อ 3 หากไม่มีเส้นทางที่สั้นกว่าให้ทำซ้ำข้อที่ 2-5 จนกว่าจะครบทุกตำแหน่ง

## 2.9.2 หลักการของ 2-Opt

หลักการ 2-Opt ถูกนำเสนอครั้งแรกโดย Croes ในปี ค.ศ.1985 เพื่อใช้ในการแก้ปัญหา TSP ซึ่งหลักการของ 2-Opt คือการหาเส้นทางทางที่สั้นในระบบย่อยใหม่ โดย

การตัดเส้นทางที่เชื่อมระหว่างจุด 2 จุด ทั้งหมด 2 เส้นทาง หลังจากนั้นทำการเชื่อมเส้นทางที่เป็นไปได้ใหม่ดังแสดงในรูปที่ 2.37 หากเส้นทางใหม่นั้นสั้นกว่าให้เก็บรูปแบบทางเดินนั้นไว้ จากนั้นก็ทำการตัดและเชื่อมต่อเส้นทางใหม่ไปเรื่อยๆ จนกว่าความยาวของเส้นทางของปัญหา TSP จะไม่มีการเปลี่ยนแปลงตามเงื่อนไขที่กำหนด เพียงเท่านี้ก็จะได้คำตอบของระบบใหญ่ของปัญหา TSP



รูปที่ 2.34 การตัดเส้นทาง และเชื่อมเส้นทางใหม่ของ 2-Opt

## 2.10 ความรู้ทั่วไปสำหรับภาษา C#

C# คือ ภาษาที่ใช้ในการเขียนโปรแกรมประเภทหนึ่งเรียกว่า Object-Oriented Programming ซึ่งถูกพัฒนาโดยบริษัทไมโครซอฟท์ โดยมีรากฐานมาจากภาษา C++ และภาษาอื่นๆ C# เป็นภาษามาตรฐานที่ถูกรองรับโดย ECAM (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) ในเดือนธันวาคม คศ. 2001 ในชื่อ ECMA-334 C# Language Specification และ ISO มาตรฐาน ISO/IEC 23270:2003 ใน คศ. 2003 ภาษา C# เป็นภาษาที่เน้นขึ้นส่วนและถูกออกแบบมาเป็นอย่างดี สามารถนำมาใช้ต่อกันเป็นอะไรก็ได้ สิ่งต่างๆ ใน ภาษา C# เป็นออบเจกต์ทั้งหมด อีกทั้งยังทนทานต่อความผิดพลาด ไม่ทำให้ระบบแฮกหรือทำงานช้าเพราะ C# มีข้อดีคือ Garbage Collection, Extension, Type-safety และ Versioning ภาษา C# เตรียมกลไกไว้หลายอย่างที่จะช่วยให้ผู้เขียนโปรแกรมสามารถนำโค้ดที่เขียนไว้ในโปรเจกต์ ไปใช้ในโปรเจกต์อื่นได้ง่าย นอกจากนั้น C# ยัง



สามารถเรียกใช้คลาสหลายพันคลาสใน .Net Framework ได้โดยตรงทำให้ลดเวลาการพัฒนาซอฟต์แวร์ได้มาก อีกทั้งยังสามารถคอมไพล์เป็น DLL ไฟล์สำหรับใช้กับภาษาอื่นได้เช่นกัน

## 2.11 ความรู้ทั่วไปสำหรับ DLL

DLL หรือ Dynamic Link Library คือ ไบรารีที่ประกอบด้วยโค้ดที่สามารถเรียกใช้งานได้มากกว่าหนึ่งโปรแกรมในเวลาเดียวกัน ข้อดีของ DLL คือโค้ดจะถูกโหลดขึ้นมาเมื่อมีการเรียกใช้งานเท่านั้น ซึ่งจะช่วงส่งเสริมให้สามารถใช้หน่วยความจำได้อย่างมีประสิทธิภาพ นอกจากนี้ยังง่ายต่อการนำไปใช้ในแต่ละโมดูล โดยไม่มีผลกระทบกับส่วนอื่นๆ ของโปรแกรม ทำให้โปรแกรมสามารถทำงานได้อย่างมีประสิทธิภาพ นอกจากนี้แล้ว DLL ยังสามารถช่วยพัฒนาโปรแกรมขนาดใหญ่ที่จำเป็นต้องมีการใช้หลายภาษาในการพัฒนาได้อีกด้วย



## บทที่ 3

### วิธีดำเนินการวิจัย

#### 3.1 กล่าวนำ

งานวิจัยนี้ได้ทำการออกแบบอัลกอริทึม สำหรับการหาเส้นทางการเดินทางที่สั้นสำหรับเครื่องจักรต้นแบบ ซึ่งมีจุดเด่นคือการใช้ระบบการมองเห็นของเครื่องจักร โดยการหาตำแหน่งสำหรับการเคลื่อนที่จะใช้ระบบการมองเห็นของเครื่องจักรในการตรวจหา และใช้วิธีการแก้ปัญหาของพนักงานขาย (TSP) ในการหาเส้นทางที่สั้นที่สุดสำหรับการเคลื่อนที่ไปยังตำแหน่งต่างๆ

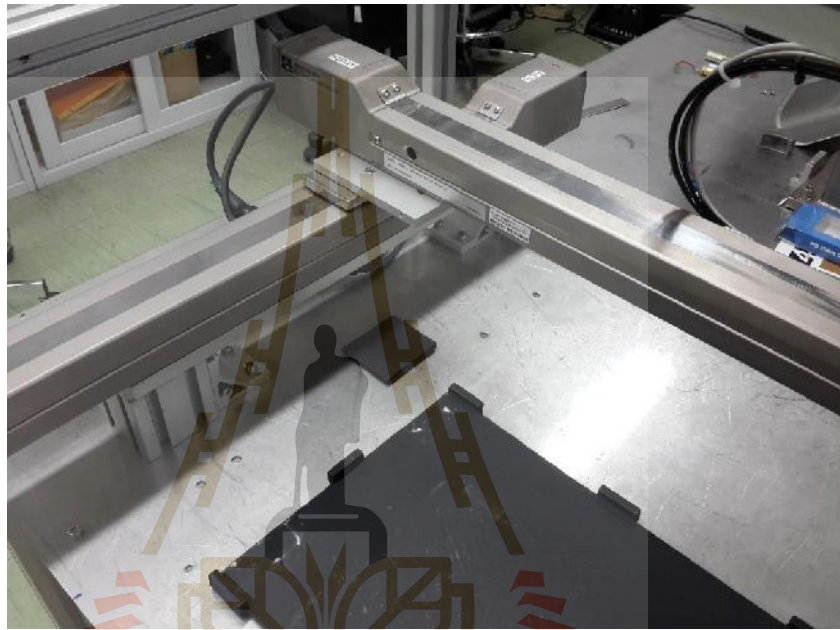
#### 3.2 อุปกรณ์ที่สำคัญสำหรับการทำวิจัย

##### 3.2.1 Cartesians Robot

Cartesian Robot คือ ใบบอทหนึ่งแกันที่เคลื่อนที่เป็นเชิงเส้น เป็นที่นิยมใช้ในโรงงานอุตสาหกรรม มีทั้งแบบ 1 แกัน 2 แกัน และ 3 แกันตามลักษณะการใช้งาน เนื่องจากเคลื่อนที่เป็นเส้นตรง มีความแม่นยำสูง มีส่วนประกอบไม่ซับซ้อน และสามารถประยุกต์ใช้งานหลากหลาย ซึ่งในงานวิจัยนี้ จะใช้ Cartesian Robot เป็นส่วนสำคัญในการเคลื่อนที่ในแนวแกน x และแกน y โดยผู้วิจัยเลือกใช้ Linear Actuator จำนวน 2 ตัว ของบริษัท IAI รุ่น RCP2CR-SA6C-I-42P-12-400P-P1-M-ESD-SP เพื่อประกอบเป็น Cartesians ใบบอทดังรูปที่ 3.1 โดยมีรายละเอียดของ Linear Actuator ดังนี้

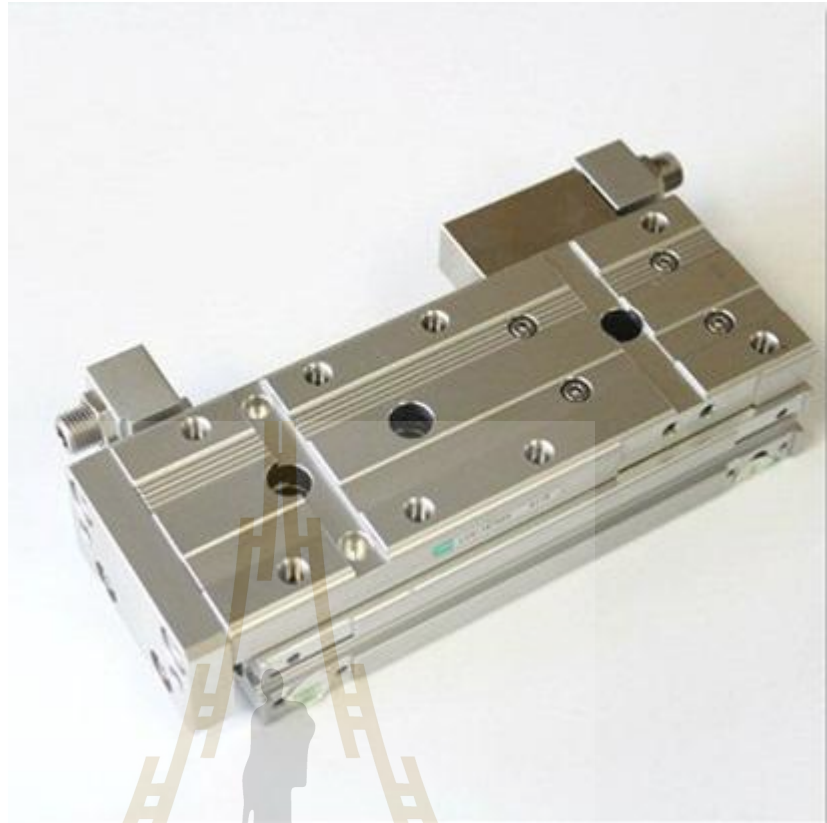
- |          |  |
|----------|--|
| - RCP2CR | หมายถึง Series ของใบบอท                        |
| - SA6C   | หมายถึงประเภทของใบบอทเป็นใบบอทสำหรับคลีนรูม    |
| - I      | หมายถึง Encoder เป็นประเภท Absolute Encoder    |
| - 42P    | หมายถึงลักษณะของมอเตอร์เป็นมอเตอร์แบบ 42 Pulse |
| - 12     | หมายถึงระยะทางเมื่อแกันของ Actuator หมุนครบ 1  |

รอบจะมีระยะการเคลื่อนที่เท่ากับ 12 มิลลิเมตร



มหาวิทยาลัยเทคโนโลยีสุรนารี

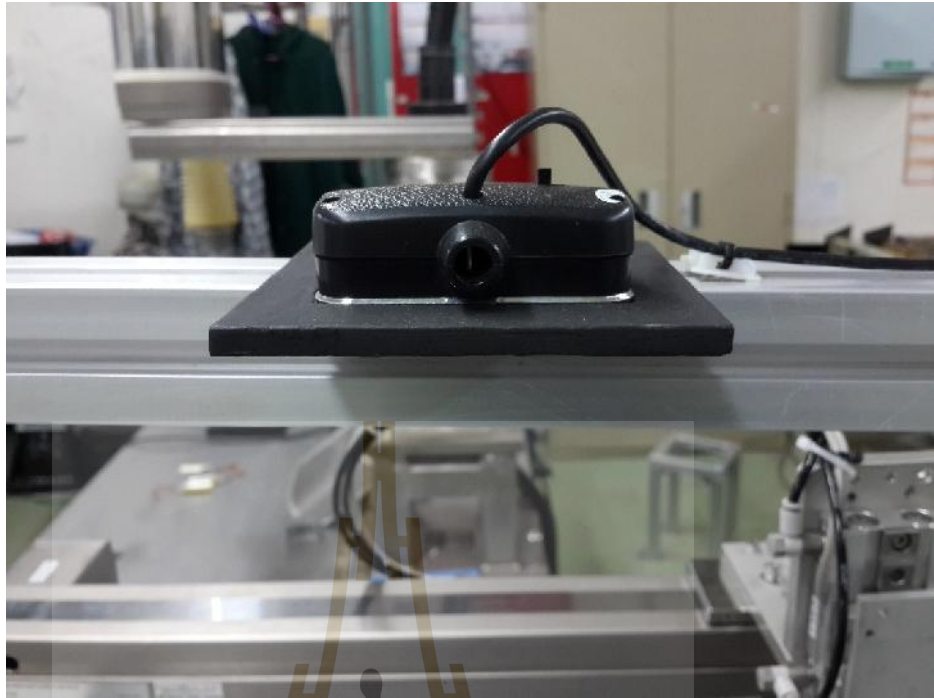




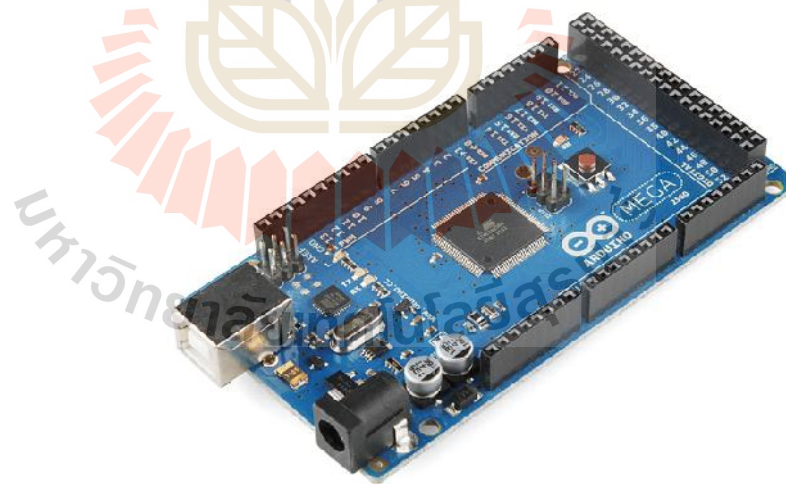




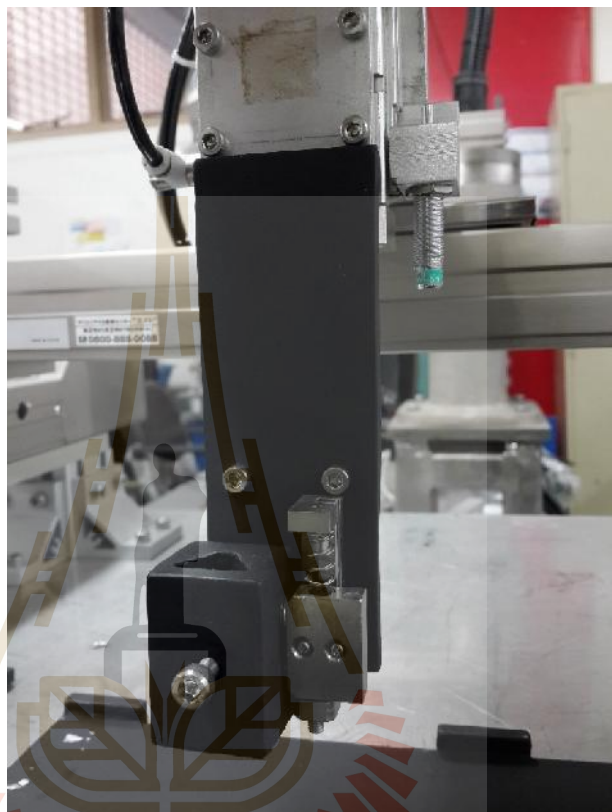




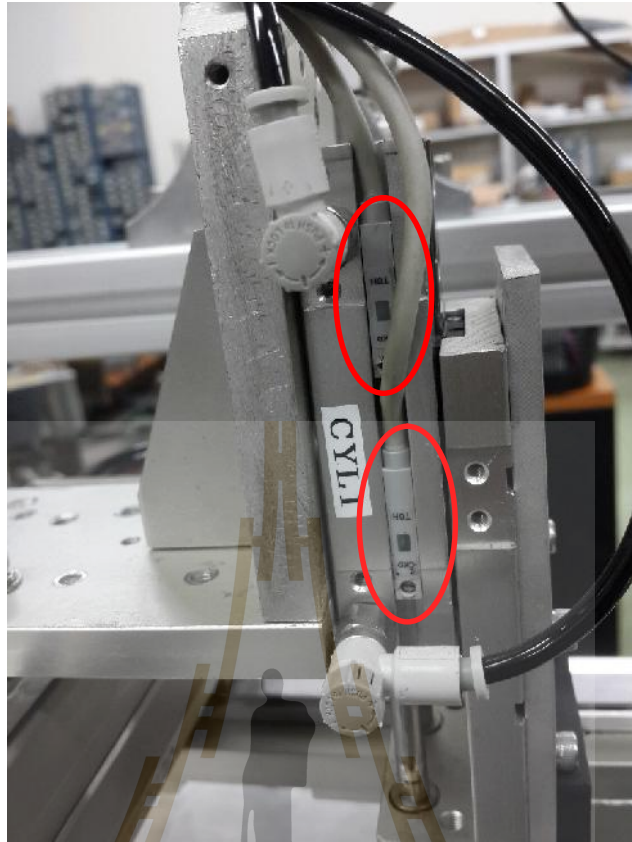








มหาวิทยาลัยเทคโนโลยีสุรนารี



### 3.3.1 ระบบการมองเห็นของเครื่องจักร

การออกแบบการมองเห็นของเครื่องจักรนั้นผู้วิจัยใช้โปรแกรม LabVIEW และ Module LabVIEW Vision ในการออกแบบระบบการมองเห็นของเครื่องจักร สามารถแบ่งออกเป็น 4 ส่วนดังนี้

- การปรับปรุงคุณภาพของภาพ
- การปรับความผิดเพี้ยนของภาพ
- การหาตำแหน่งของวงกลมภายในภาพ
- การปรับระบบโคออร์ดิเนตของภาพให้เป็นโคออร์ดิเนตของโลกจริง

### 3.3.2 การควบคุม Cartesian Robot

การควบคุมโรบอทนั้นสามารถควบคุมด้วยโปรแกรมโดยผ่าน Hardware 2 ส่วน คือ กล้องควบคุมโรบอทของบริษัท IAI และการควบคุมผ่าน ไมโครคอนโทรลเลอร์ของ Arduino โดยการออกแบบนั้น ผู้วิจัยได้ทำการเขียนเป็น Module หรือ Sub VI สำหรับการควบคุมซึ่งประกอบด้วย Module ต่างๆ ดังนี้

- Axis Status Decode ใช้สำหรับการถอดรหัสข้อมูลที่โรบอทส่งกลับมายังโปรแกรม

- Axis Status X-Y Command คือชุดคำสั่งการสอบถามสถานะปัจจุบันของโรบอท

- Axis Status บอกสถานะปัจจุบันของแกน

- Hex to Dec แปลงเลขฐาน 16 เป็นเลขฐาน 10 เนื่องจากตำแหน่งของโรบอทในชุด

ของคำสั่งที่ถูกส่งกลับมาเป็นเลขฐาน 16

- Home X-Y ชุดคำสั่งให้ทำการ Homing โรบอททั้งแกน x และแกน y

- On Off Servo ชุดคำสั่งสำหรับการใช้งานมอเตอร์ของโรบอท

- Robot Move X-Y ชุดคำสั่งสำหรับการเคลื่อนที่ไปยังตำแหน่งต่างๆ ในพิกัด x

และ y

- Robot Reset ชุดคำสั่งสำหรับ reset โรบอท

- Robot Status ใช้สำหรับบอกสถานะปัจจุบันของโรบอท

- Send Received Data ใช้สำหรับส่งและอ่านข้อมูลระหว่างโปรแกรมกับโรบอท

- Interface Arduino Read ใช้สำหรับอ่านข้อมูลจากไมโครคอนโทรลเลอร์

- Interface Arduino Write ใช้ส่ง output ไปยังไมโครคอนโทรลเลอร์

ซึ่ง Module เหล่านี้ใช้ในควบคุมโรบอทและใช้สำหรับอ่านค่าสถานะต่างๆ ของโรบอทเพื่อให้สามารถควบคุมการทำงานได้อย่างมีประสิทธิภาพ รวมทั้งยังสามารถตรวจสอบความผิดพลาดได้โดยง่าย

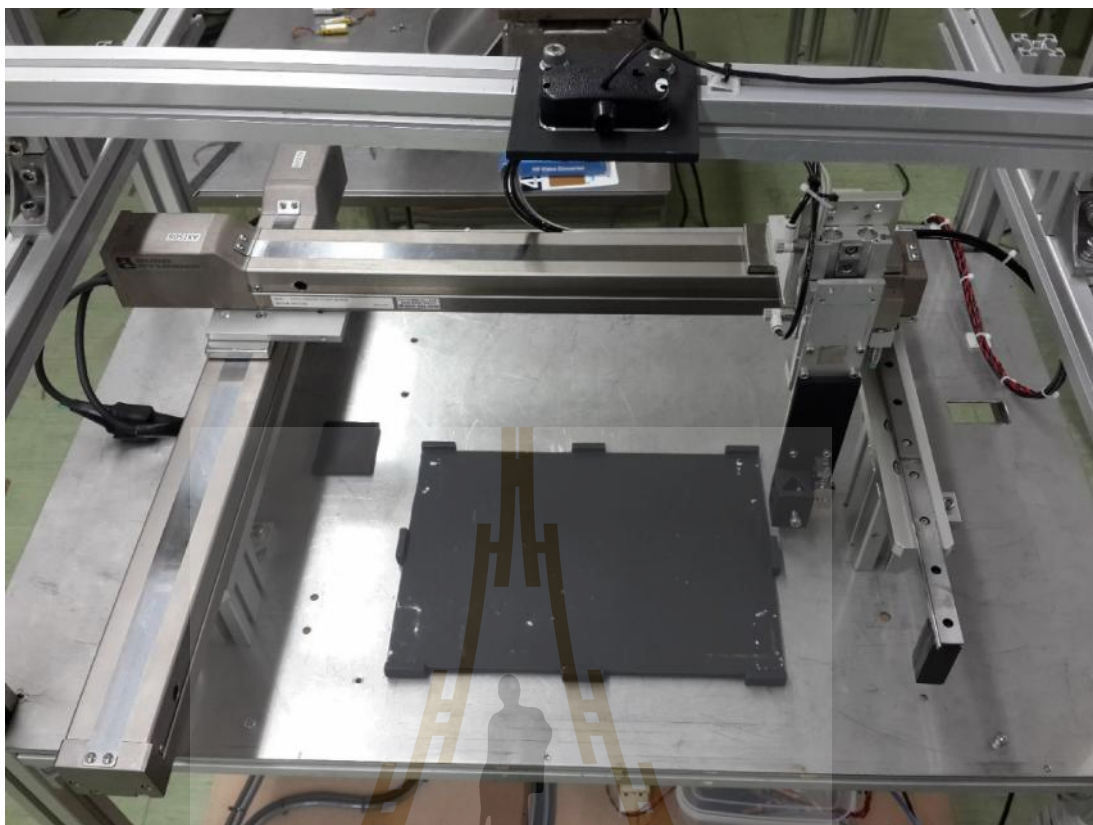
### 3.3.3 การหาเส้นทางที่สั้นที่สุดโดยใช้ TSP

การออกแบบอัลกอริทึมของการแก้ปัญหาการเดินทางของพนักงานขายนั้น มีความยุ่งยากและซับซ้อน อีกทั้งยังต้องการความรวดเร็วในการประมวลผล ด้วยเหตุนี้ผู้วิจัยจึงใช้ภาษา C# ในการเขียน Library เพื่อใช้กับโปรแกรม LabVIEW อัลกอริทึมที่ใช้ในการแก้ปัญหาการเดินทางของพนักงานขาย ซึ่งจะประกอบด้วย 2 อัลกอริทึมโดยใช้ Branch and Bound ในการหาเส้นทางเริ่มต้นซึ่งใช้ในการแก้ปัญหา TSP สำหรับการแก้ปัญหาแบบฮิวริสติกดังรายการอ้างอิงที่ 3 และหลังจากนั้นจะเป็นในส่วนของการปรับปรุงระยะทางให้สั้นลงโดยใช้ 2-Opt อัลกอริทึม ซึ่งมีผลการวิจัยการใช้อัลกอริทึมทั้งสองในการแก้ปัญหา TSP ดังรายการอ้างอิงที่ 11

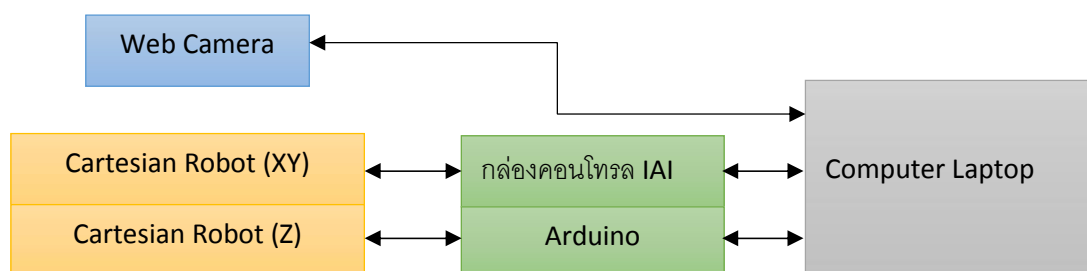
### 3.4 ขั้นตอนการทำงานของระบบ

ในงานวิจัยนี้ผู้วิจัยได้ทำการสร้างเครื่องจักรต้นแบบ เพื่อทดสอบอัลกอริทึมดังรูปที่ 3.13 โดยการใช้ Cartesian Robot ร่วมกับการมองเห็นของเครื่องจักรโดยมีการเชื่อมต่อระหว่างอุปกรณ์แต่ละส่วนดังรูปที่ 3.14 ขั้นตอนการทดสอบดังแสดงในรูปที่ 3.15 และขั้นตอนการทำงานของระบบดังรูปที่ 3.16 ซึ่งแต่ละส่วนมีรายละเอียดดังนี้

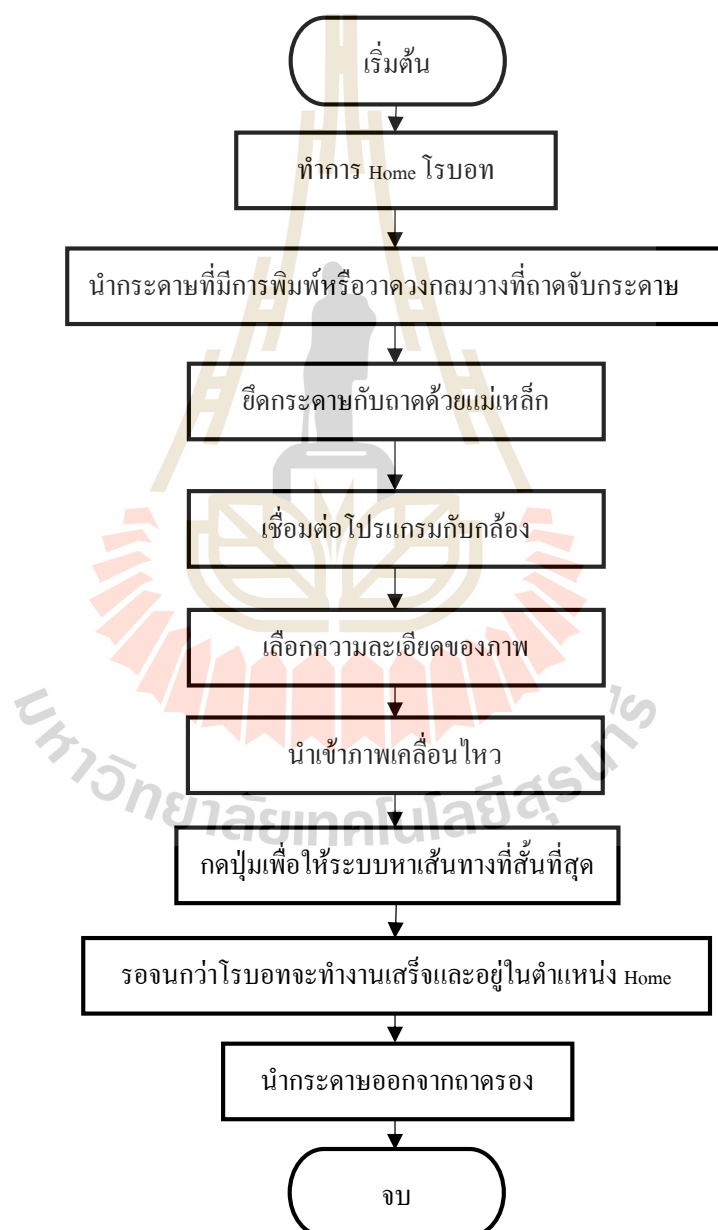
มหาวิทยาลัยเทคโนโลยีสุรนารี



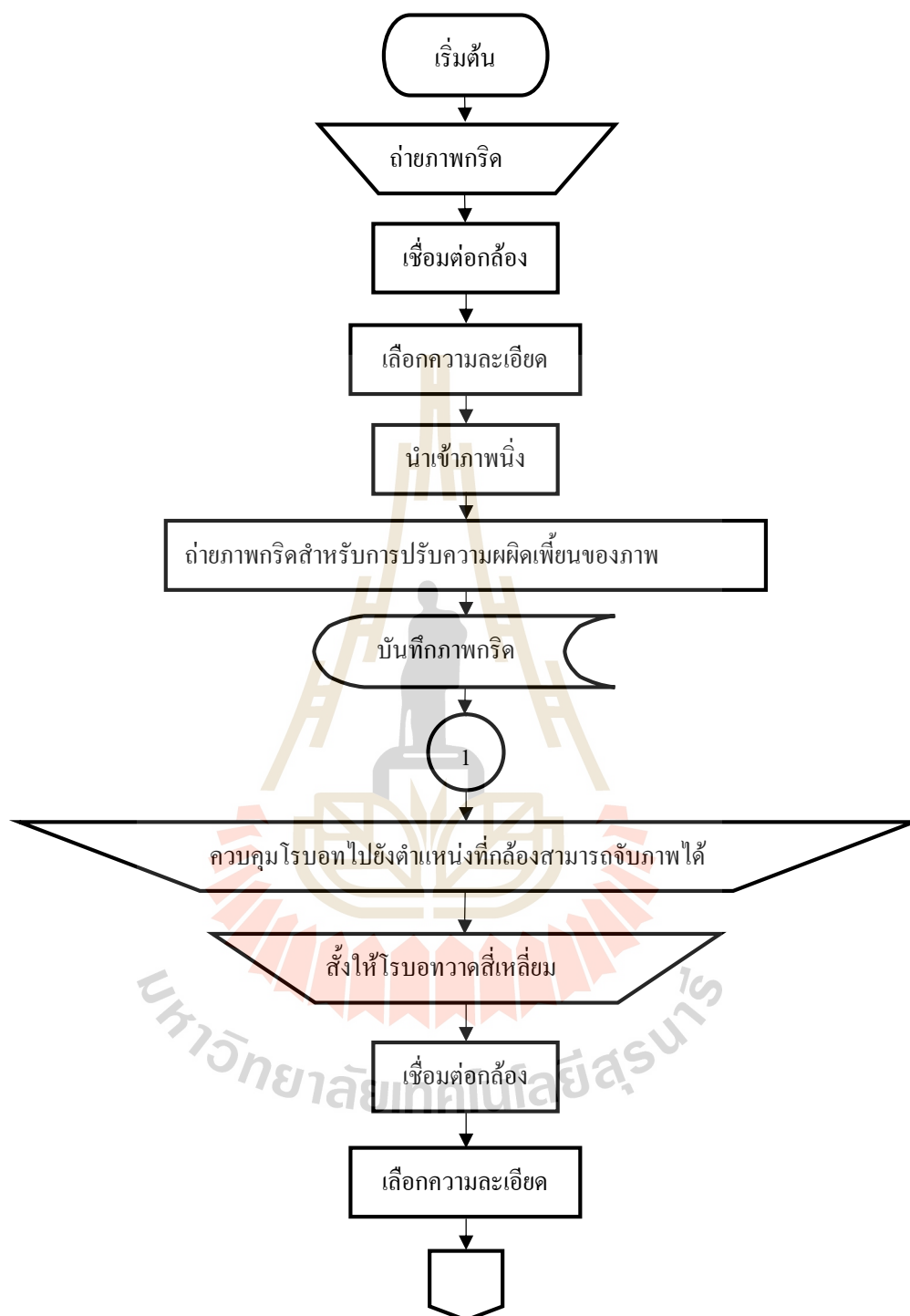




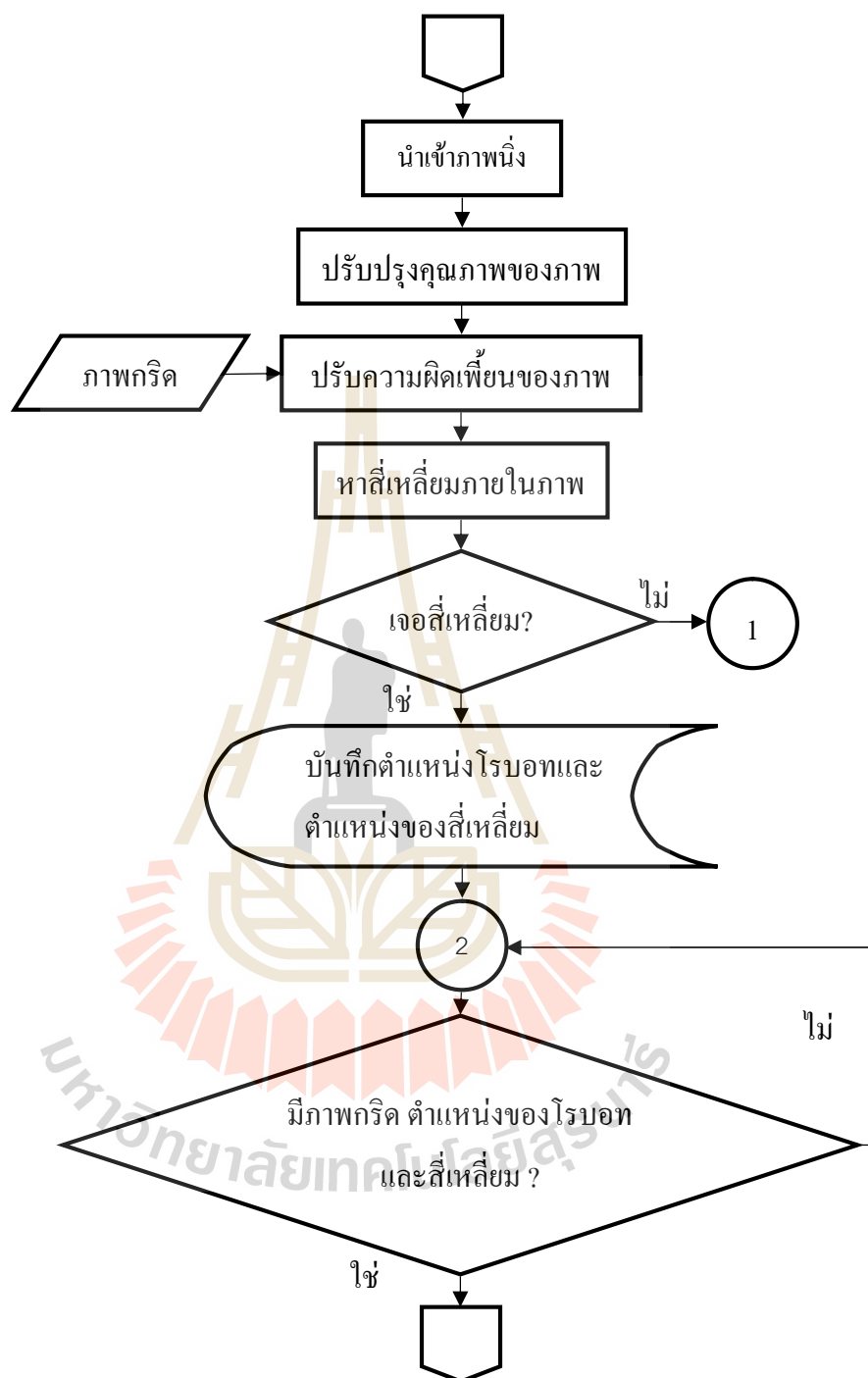
รูปที่ 3.14 การเชื่อมต่อระหว่างอุปกรณ์



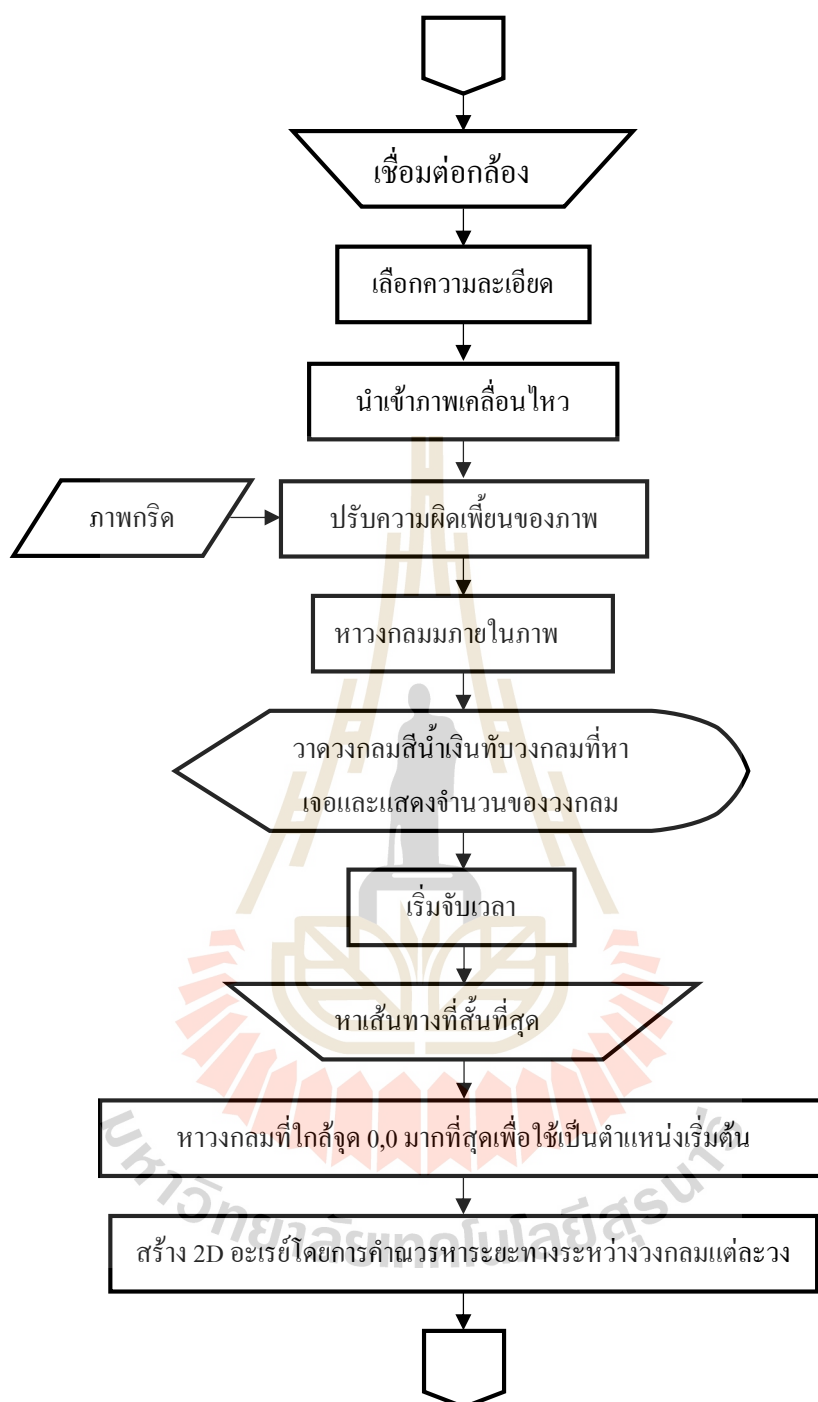
รูปที่ 3.15 ขั้นตอนการทดสอบ



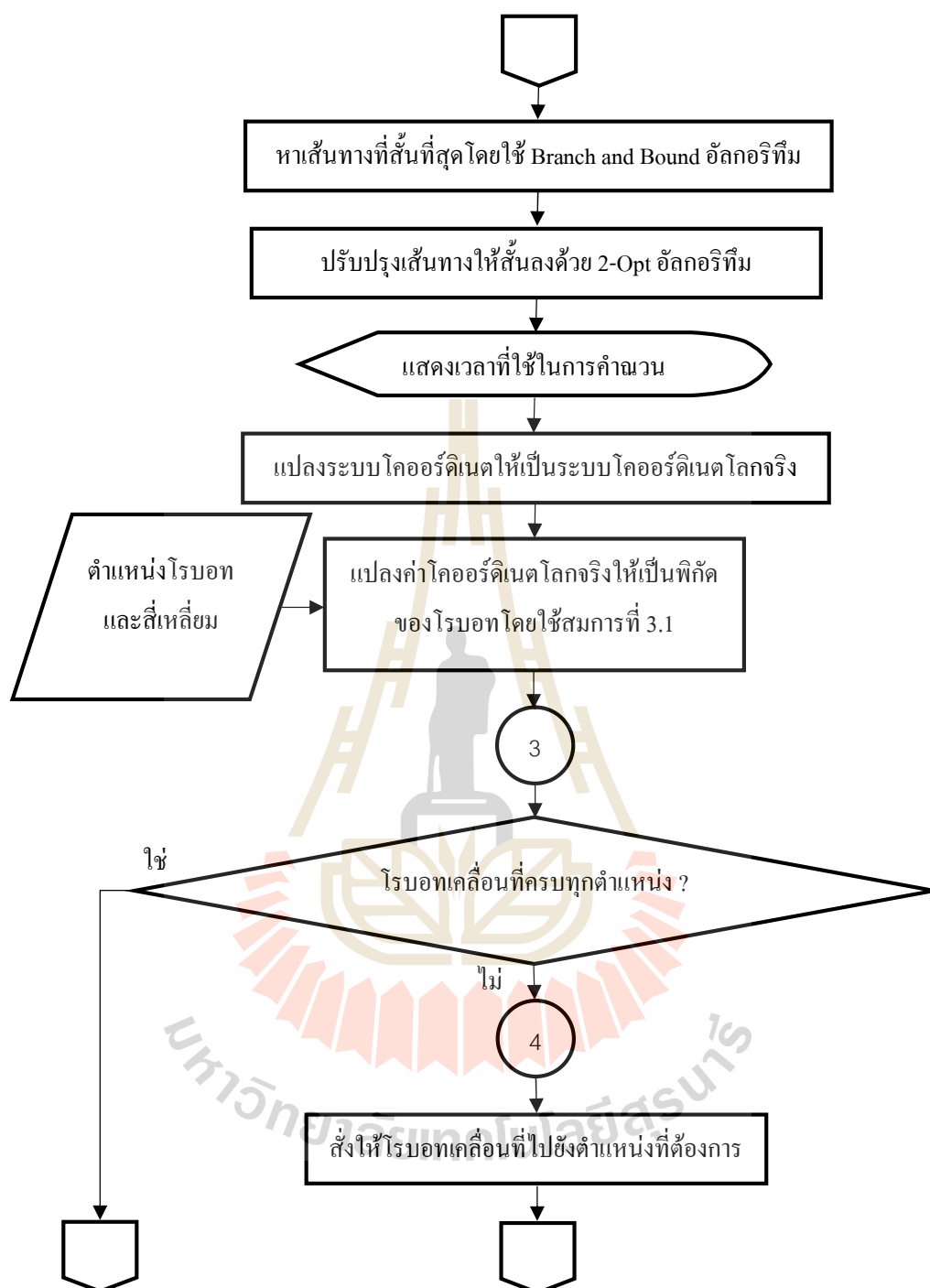
รูปที่ 3.16 ขั้นตอนการทำงานของระบบ



รูปที่ 3.17 ขั้นตอนการทำงานของระบบ (ต่อ)



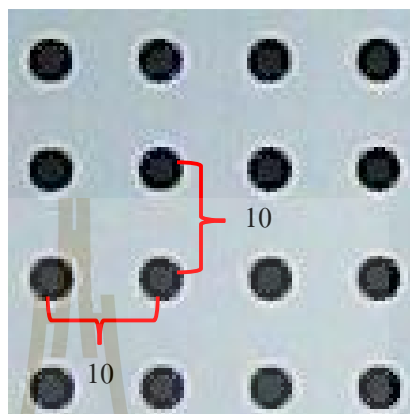
รูปที่ 3.18 ขั้นตอนการทำงานของระบบ (ต่อ)



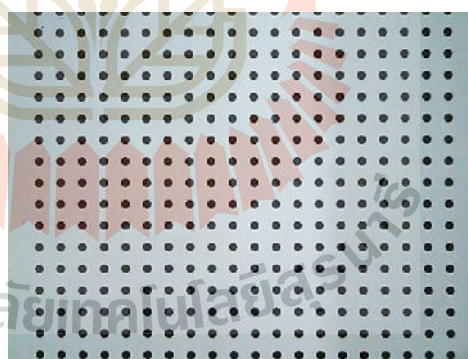
รูปที่ 3.19 ขั้นตอนการทำงานของระบบ (ต่อ)



ของภาพเท่ากับ  $640 \times 480$  พิกเซลโดยครอบคลุมพื้นที่  $203 \times 152$  มิลลิเมตร หลังจากนั้นทำการบันทึกภาพลงบนคอมพิวเตอร์ดังแสดงในรูปที่ 3.22 จากภาพที่ได้เราจะนำไปเพื่อใช้ในการปรับความผิดเพี้ยนของภาพต่อไป



รูปที่ 3.21 ระยะระหว่างวงกลมเท่ากับ 10 มิลลิเมตร ตามแกน x และ แกน y

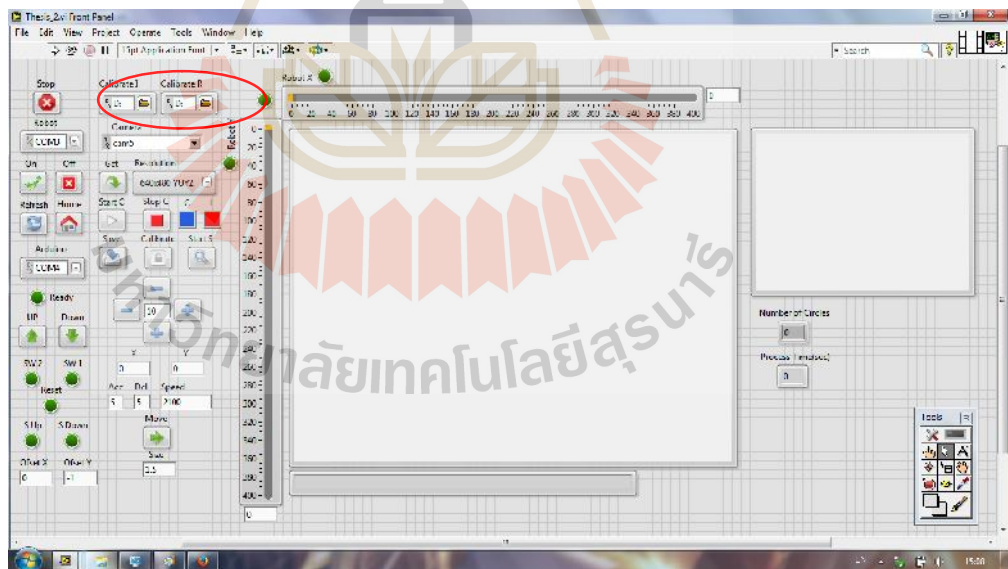


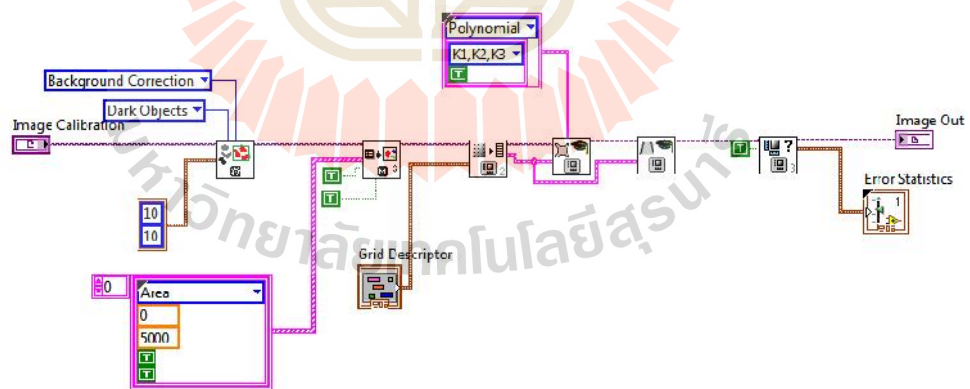
รูปที่ 3.22 ภาพกริดที่ถ่ายจากกล้อง เพื่อใช้ในการปรับความผิดเพี้ยนของภาพ

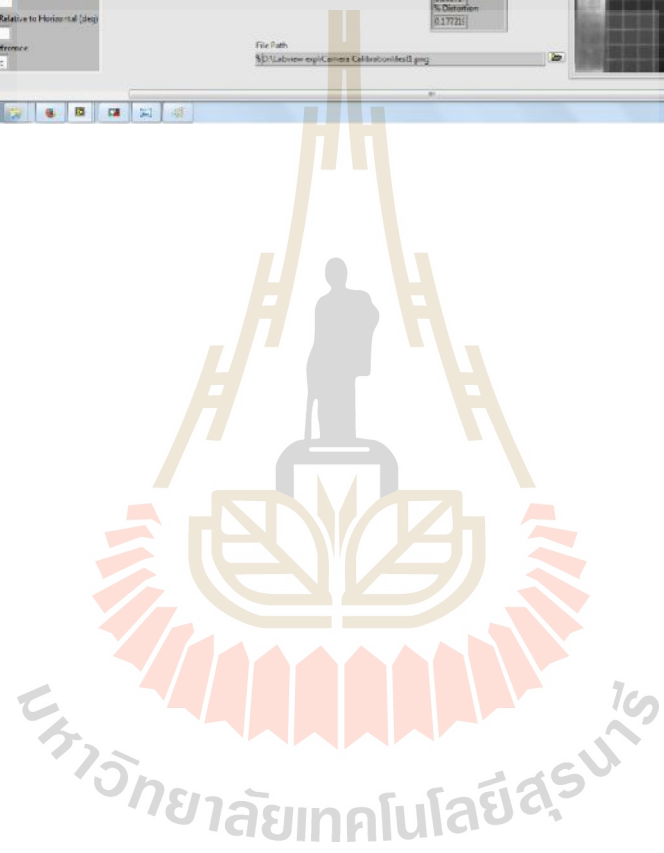
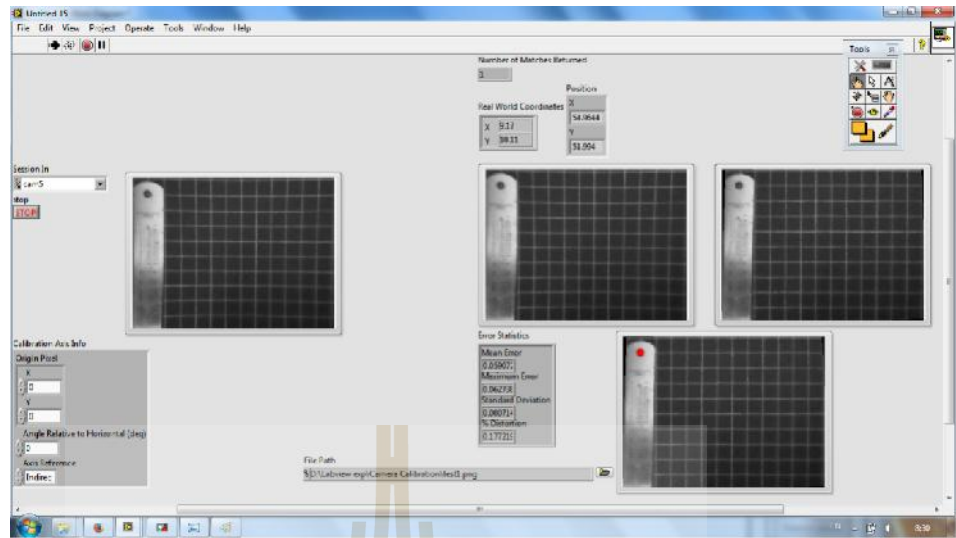
### 3.4.2 การบันทึกตำแหน่งอ้างอิงของโรบอทกับภาพที่ได้จากกล้อง

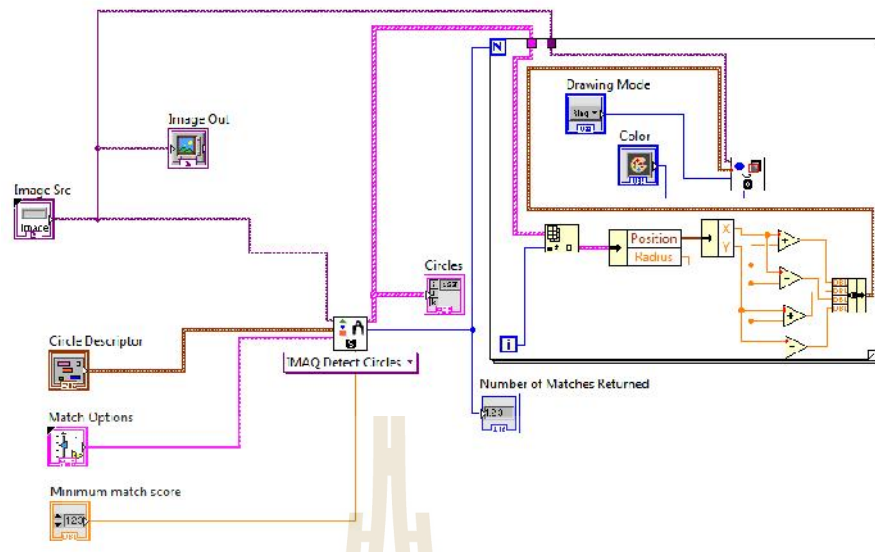
ในขั้นตอนนี้ จะต้องทำการติดตั้งปากกาไว้ที่ตัวโรบอทก่อน แล้วทำการตั้งการให้โรบอททำการลากเส้นตรง และทำการวัดขนาดของเส้นที่ลาก เพื่อใช้ในการคำนวณตำแหน่งของโรบอท หลังจากนั้นจึงให้โรบอทเคลื่อนที่ไปยังตำแหน่งที่กล้องสามารถจับภาพได้ แล้วเริ่มตั้งการให้โรบอทเคลื่อนที่เพื่อวาดรูปสี่เหลี่ยมลงบนกระดาษ โดยตำแหน่งของสี่เหลี่ยมจะเลื่อนจาก









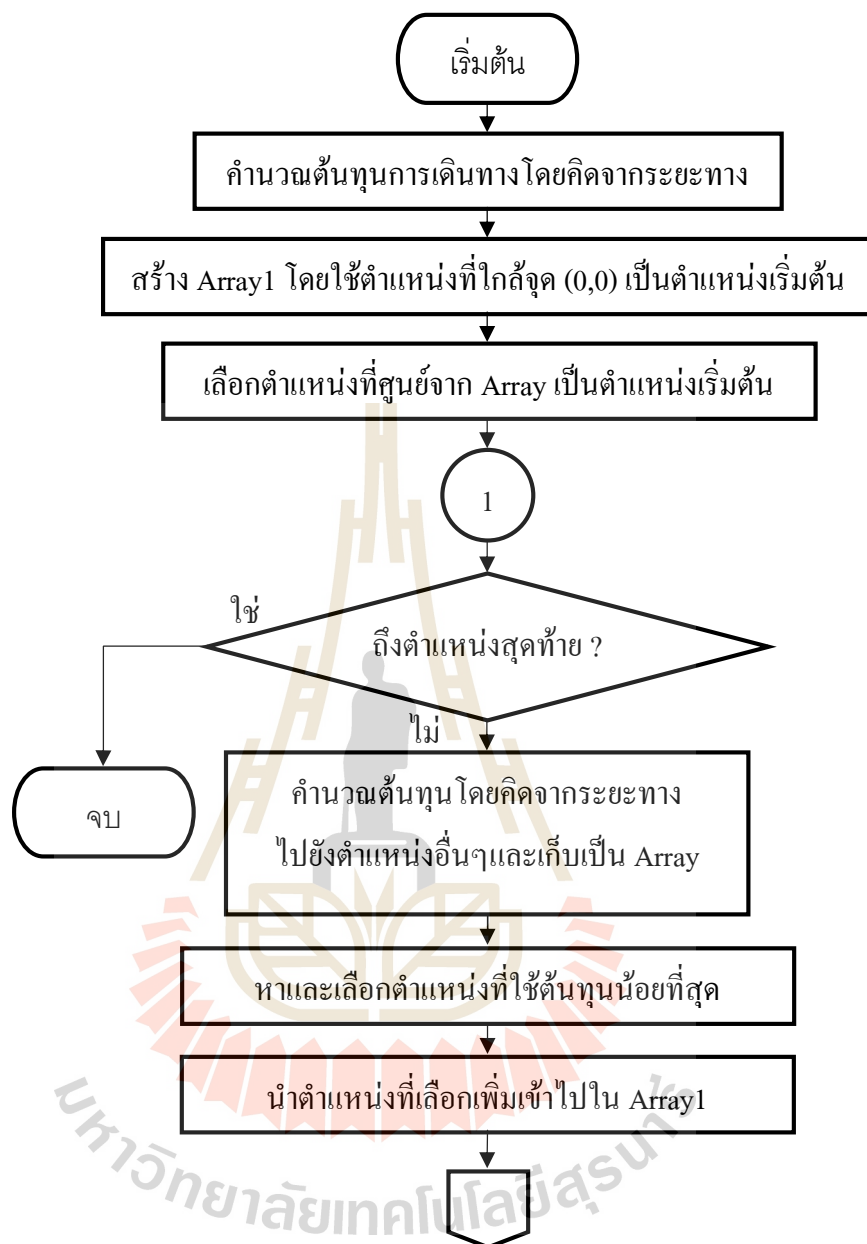


มหาวิทยาลัยเทคโนโลยีสุรนารี

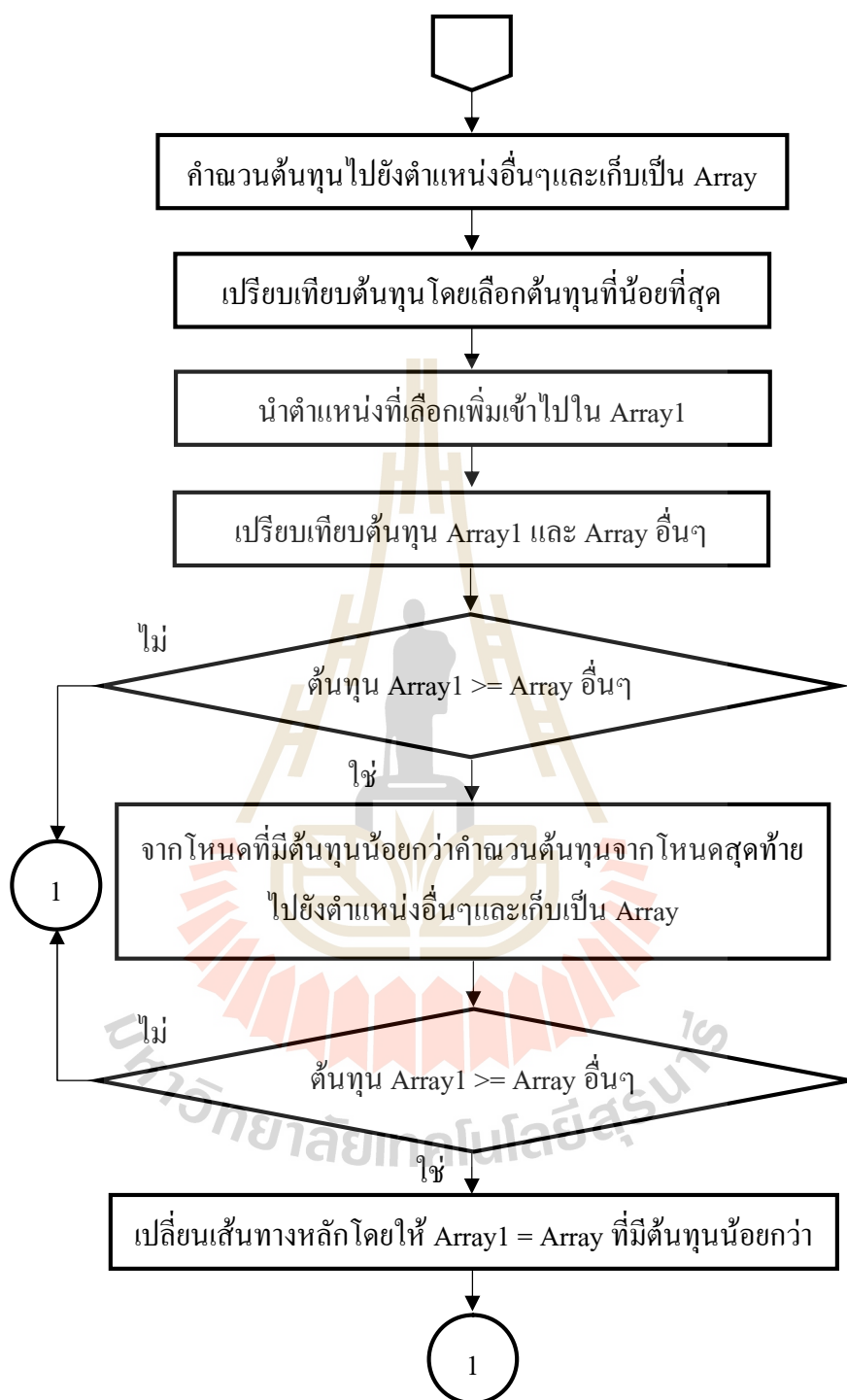
ได้เริ่มมีการพัฒนาตัวแบบทางคณิตศาสตร์และวิธีการแม่นยำตรง (exact method) Richard M. K(1972) ได้พิสูจน์ว่าเกมส่วกลมหามิทอนเนียนเป็นปัญหาประเภท NP-complete ซึ่งทำให้ปัญหา TSP เป็นปัญหายากพวกเดียวกันได้นั้นคือเป็นปัญหา NP-hardness จึงแสดงให้เห็นว่า TSP เป็นปัญหาที่ยากและยังไม่มีวิธีการที่ใช้เวลาแบบโพลิโนเมียลในการแก้ปัญหาได้ ดังนั้นนักวิจัยจึงสนใจทั้งหาวิธีในการแก้ปัญหาแบบวิธีการที่ได้คำตอบที่ดีที่สุดเช่น วิธีการ branch and bound วิธีการ branch and cut, cutting plane algorithm , column generation เป็นต้น

ในงานวิจัยนี้ใช้การแก้ปัญหาที่เรียกว่า Branch and Bound ในการหาเส้นทางที่สั้นที่สุด ดังรูปที่ 3.27 หลักการของการแก้ปัญหาคือ เริ่มต้นด้วยการหาเส้นทางขึ้นมาเส้นทางหนึ่ง แล้วพยายามหาเส้นทางอื่นขึ้นมาเปรียบเทียบ ถ้าเส้นทางใหม่ที่ได้ดีกว่า ให้นำเส้นทางใหม่ที่ได้ใช้เป็นเส้นทางหลักในการเปรียบเทียบกับเส้นทางอื่นต่อไป แต่ถ้าเส้นทางใหม่ที่สร้างขึ้น เมื่อคำนวณแล้ว พบว่าแม้เพียงบางส่วนที่ยาวกว่าเส้นทางหลัก ก็ให้ยกเลิกเส้นทางนั้น ทำเช่นนี้เรื่อยๆจนกระทั่งได้เส้นทางที่ดีที่สุด เพื่อใช้เป็นรูปแบบการเดินทางเริ่มต้น หลังจากนั้นใช้ 2-Opt อัลกอริทึมทำการหา Local minimum เพื่อปรับปรุงคุณภาพของเส้นทางให้มีเส้นทางที่สั้นลงจนกว่าอัลกอริทึมจะไม่สามารถปรับปรุงความยาวได้ดังรูปที่ 3.28 ก็จะได้อัลกอริทึมที่สั้นที่สุดในการเคลื่อนที่ไปยังตำแหน่งต่างๆ ในขั้นตอนนี้ผู้วิจัยใช้ภาษา C# ในการเขียน Library ของ Branch and Bound และ 2-Opt จากนั้นแปลง Library เป็นไฟล์ .dll เพื่อใช้ในโปรแกรม LabVIEW



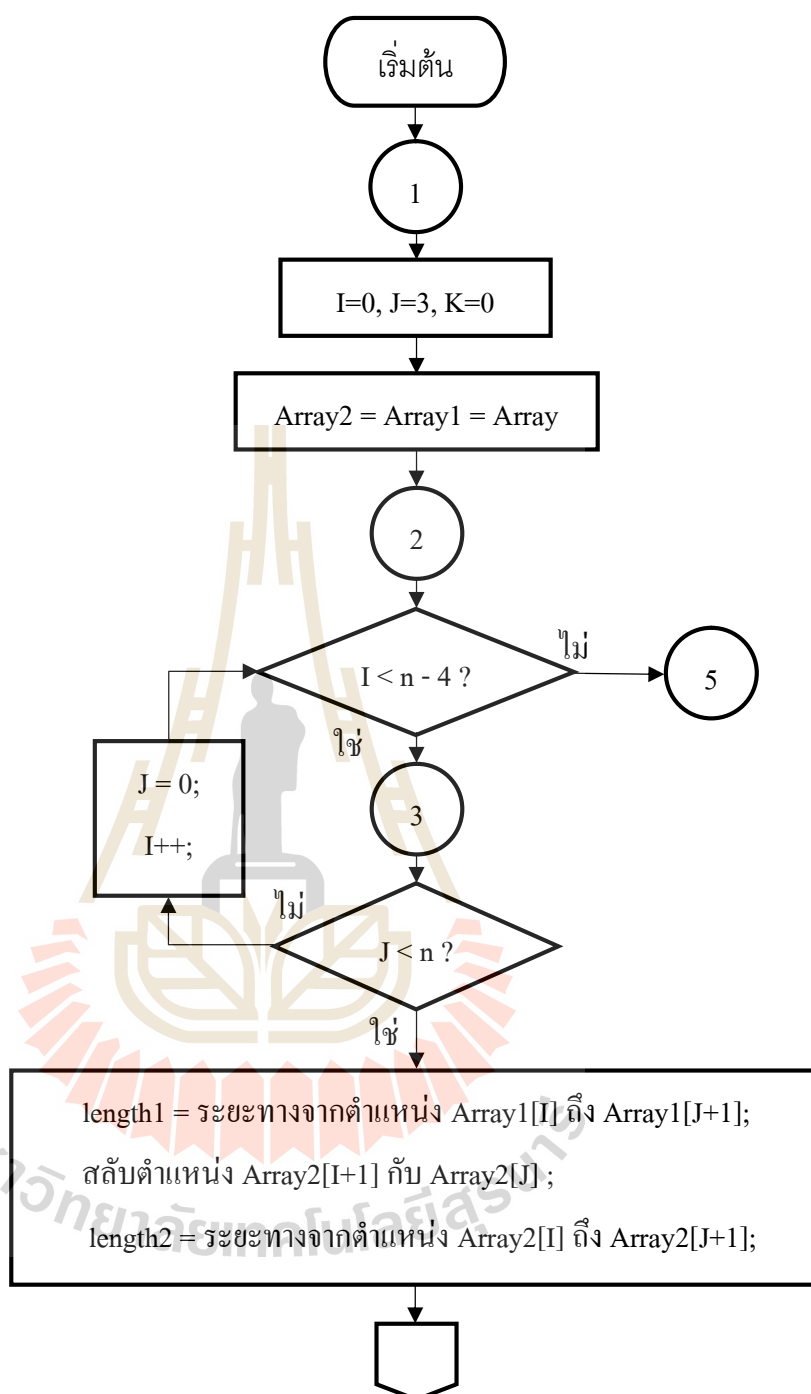


รูปที่ 3.27 ขั้นตอนการหาเส้นทางโดยใช้ Branch and bound

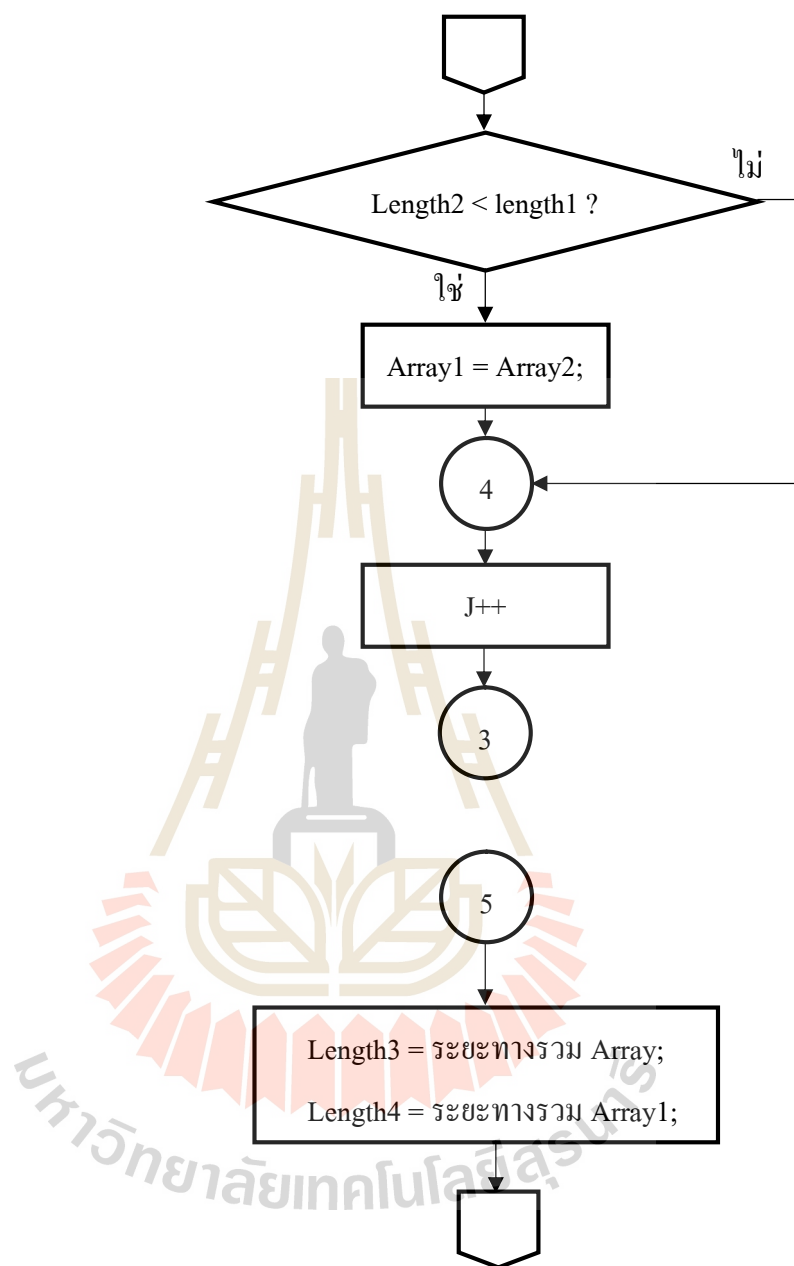


รูปที่ 3.28 ขั้นตอนการหาเส้นทางโดยใช้ Branch and bound (ต่อ)

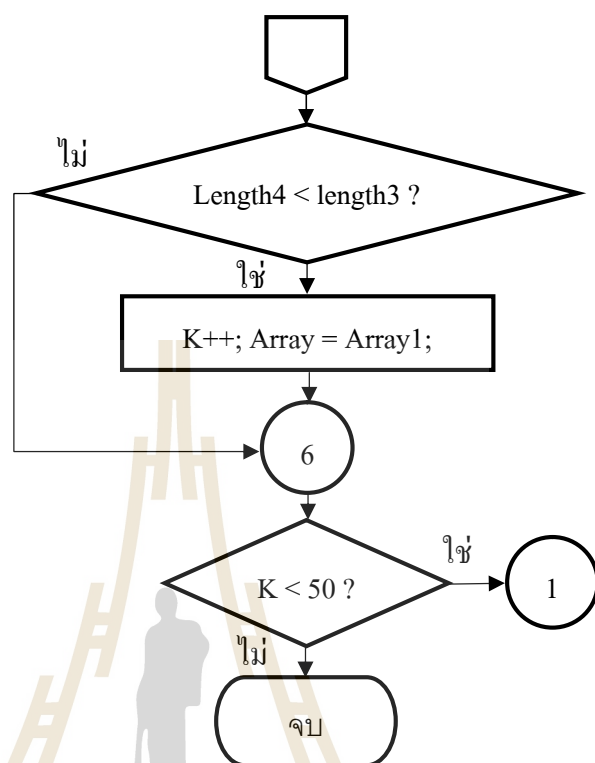




รูปที่ 3.29 ขั้นตอนการปรับปรุงเส้นทางให้สั้นลงโดยใช้ 2-Opt โดยที่  $n$  คือจำนวนจุดทั้งหมด



รูปที่ 3.30 ขั้นตอนการปรับปรุงเส้นทางให้สั้นลงโดยใช้ 2-Opt โดยที่  $n$  คือจำนวนจุดทั้งหมด(ต่อ)



รูปที่ 3.31 ขั้นตอนการปรับปรุงเส้นทางให้สั้นลงโดยใช้ 2-Opt โดยที่ n คือจำนวนจุดทั้งหมด(ต่อ)

### 3.4.8 ทำการแปลงระบบโคออร์ดิเนต

การแปลงระบบโคออร์ดิเนตนั้นผู้วิจัยทำการเขียน module ที่ประกอบไปด้วยการคำนวณการปรับระยะการเลื่อนของภาพและการคำนวณการหมุนของภาพ หลังจากนั้นจึงทำการแปลงระบบโคออร์ดิเนตโดยใช้ module สำเร็จของ LabVIEW Vision และนำค่าที่ได้จากการแปลงระบบโคออร์ดิเนตมาใช้ในการคำนวณสำหรับตำแหน่งใดๆ ของโรบอท ดังสมการต่อไปนี้

$$Rb_{(x,y)} = R_{(x,y)} + (P_{(x,y)} - I_{(x,y)}) \quad (3.1)$$

โดยที่

$Rb_{(x,y)}$  คือตำแหน่งของใดๆ ของโรบอท

$R(x,y)$  คือตำแหน่งซ้ายบนที่โรบอทวาดสี่เหลี่ยมในหัวข้อที่ 3.4.2

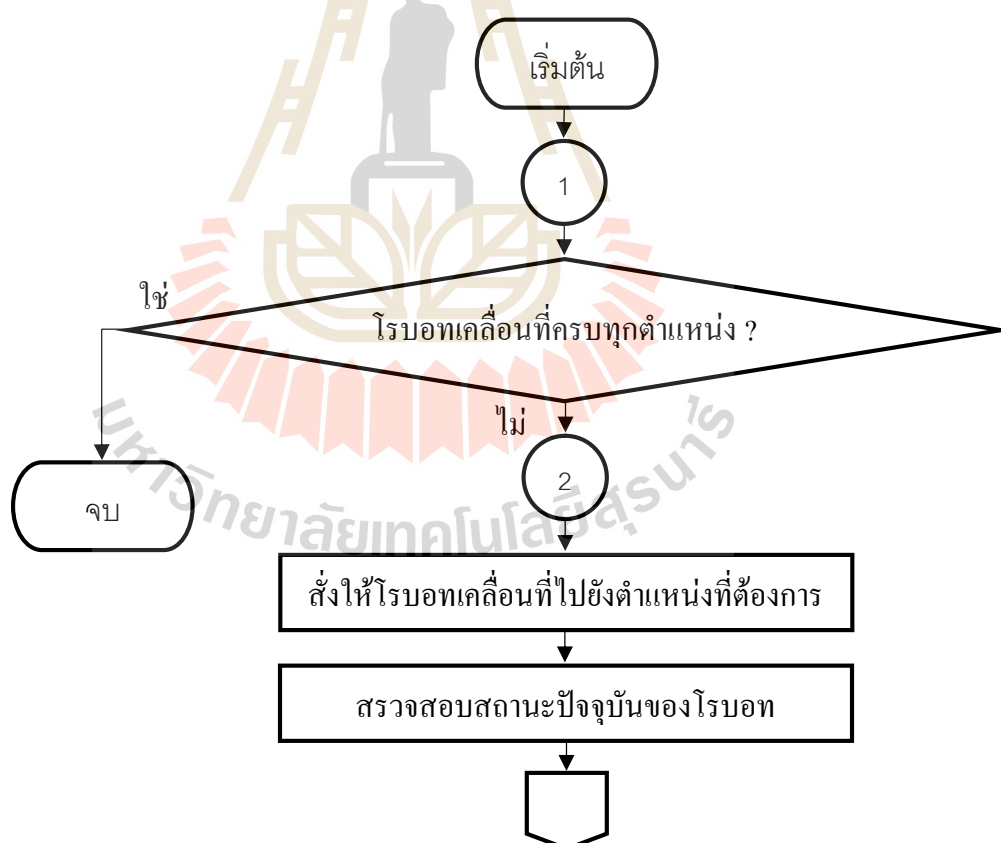
$P(x,y)$  คือตำแหน่งใดๆบนภาพที่ทำการแปลงโคออร์ดิเนตแล้ว

$I(x,y)$  คือตำแหน่งซ้ายบนของสี่เหลี่ยมที่พบในภาพของหัวข้อที่ 3.4.2

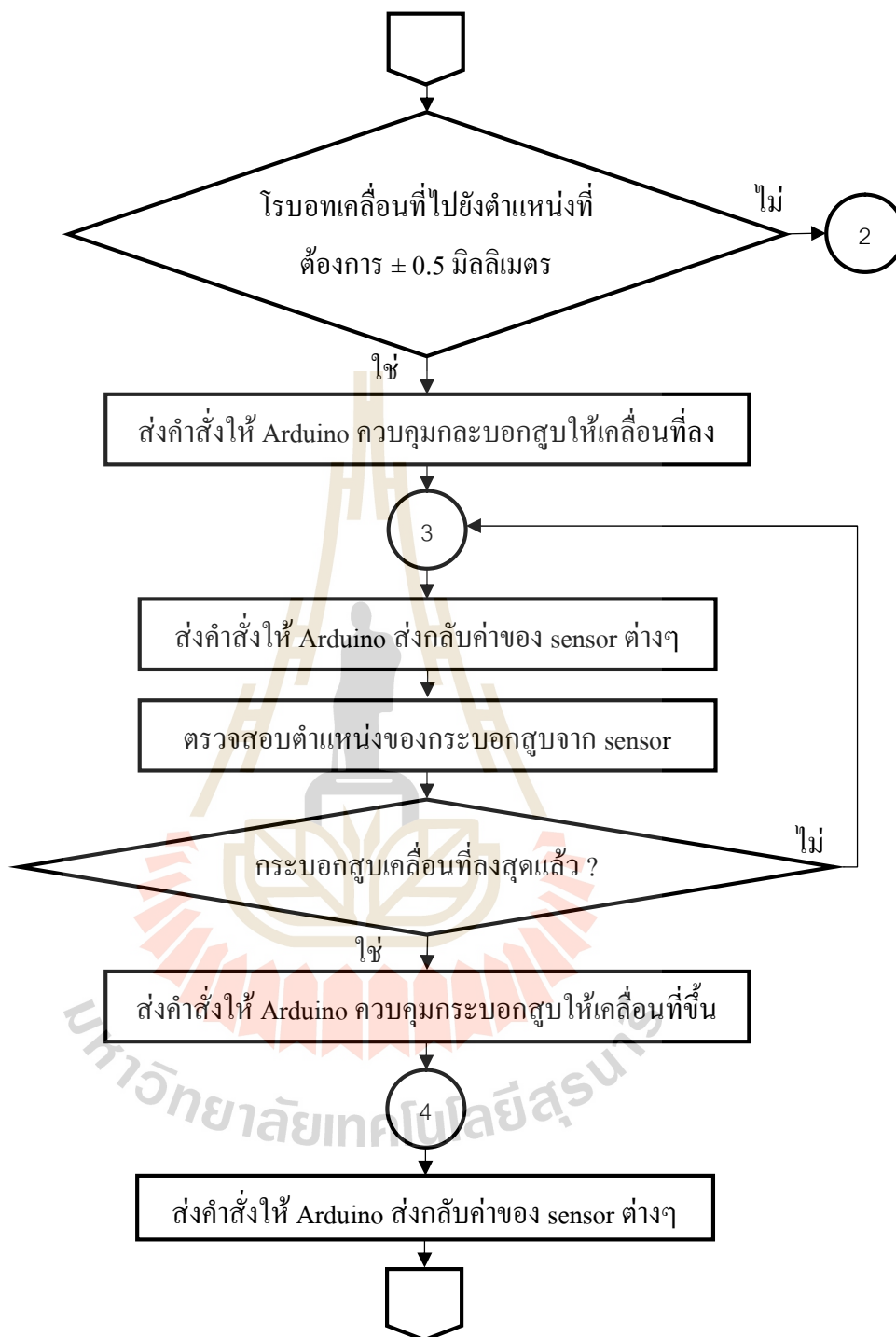
ซึ่งตำแหน่งต่างๆ ที่คำนวณแล้ว ข้อมูลจะถูกเก็บอยู่ในรูปแบบของอะเรย์ ก่อนที่จะควบคุมให้โรบอทเคลื่อนที่ไปยังตำแหน่งต่างๆ

### 3.4.9 ควบคุมโรบอทให้เคลื่อนที่ไปยังตำแหน่งต่างๆ

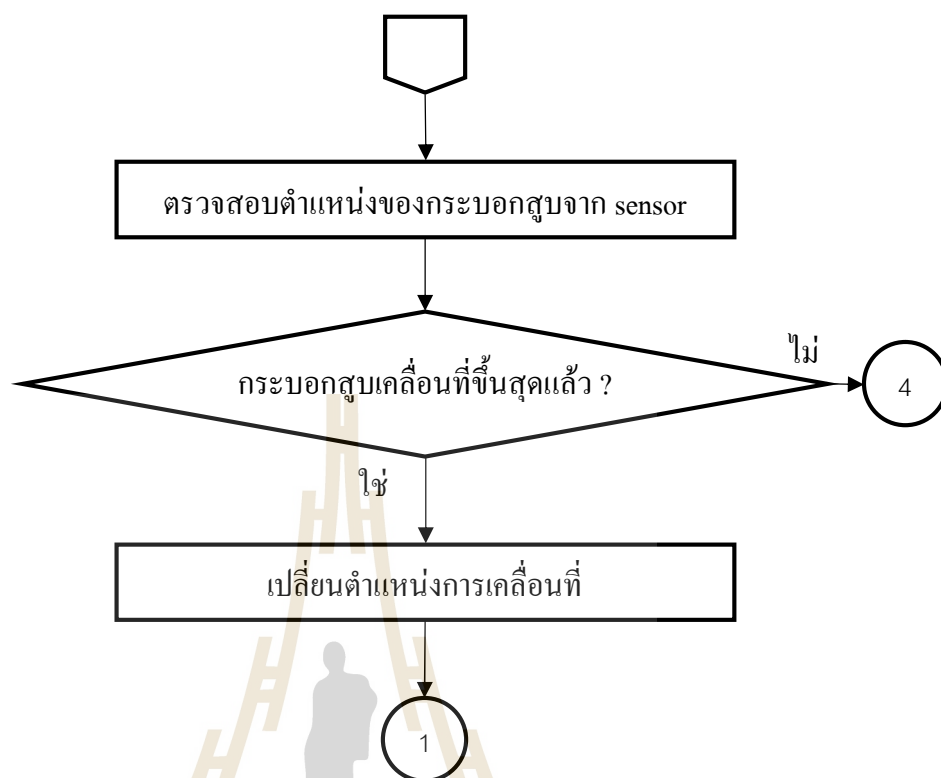
การทำงานในส่วนนี้คือการควบคุมให้โรบอทเคลื่อนที่ไปยังตำแหน่งต่างๆและการควบคุมกระบอกสูบด้วยไมโครคอนโทรลเลอร์ ให้ทำงานสัมพันธ์กัน ซึ่งถูกควบคุมโดยโปรแกรม ซึ่งการทำงานของโปรแกรมจะเป็นแบบ Close loop โปรแกรมจะมีการสื่อสารระหว่างโรบอทและไมโครคอนโทรลเลอร์ตลอดเวลา เพื่อยืนยันตำแหน่งปัจจุบันของโรบอทและตำแหน่งกระบอกสูบซึ่งมีขั้นตอนการทำงานดังรูปที่ 3.3



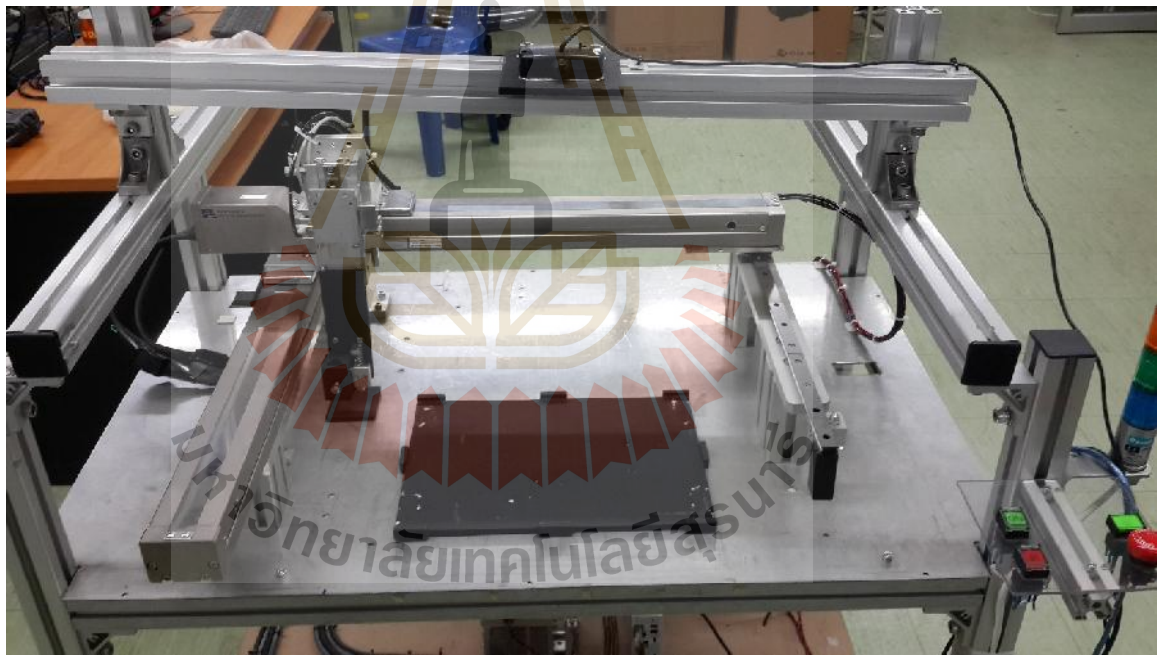
รูปที่ 3.32 การควบคุมโรบอทไปยังตำแหน่งต่างๆ



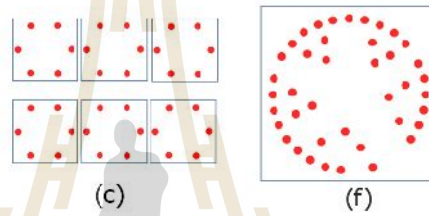
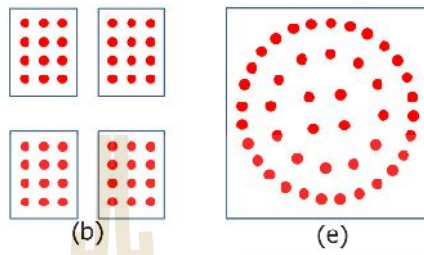
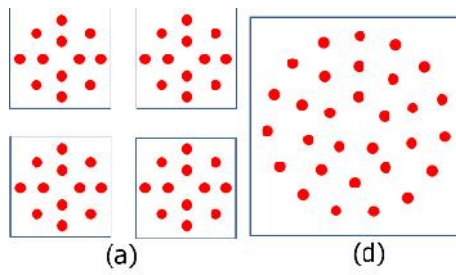
รูปที่ 3.33 การควบคุมrobotไปยังตำแหน่งต่างๆ (ต่อ)



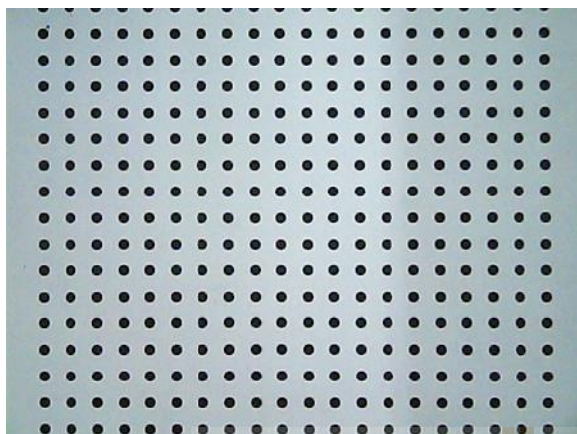
รูปที่ 3.34 การควบคุม 로봇ไปยังตำแหน่งต่างๆ (ต่อ)



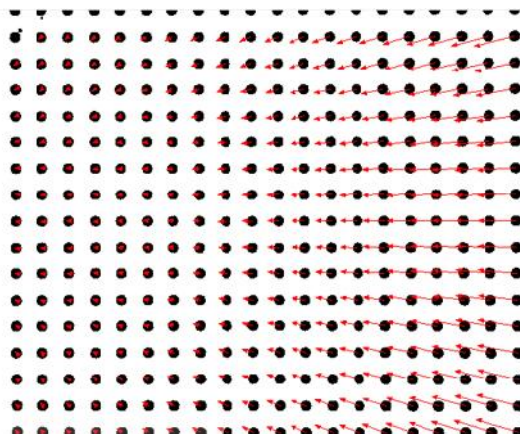




มหาวิทยาลัยเทคโนโลยีสุรนารี

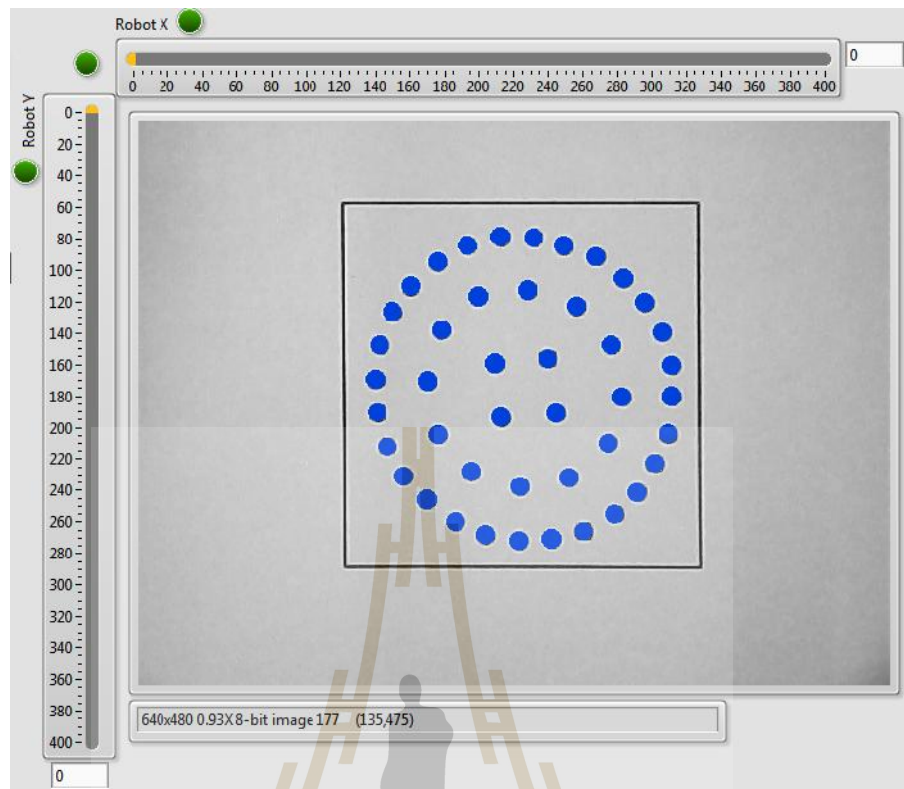


(a)

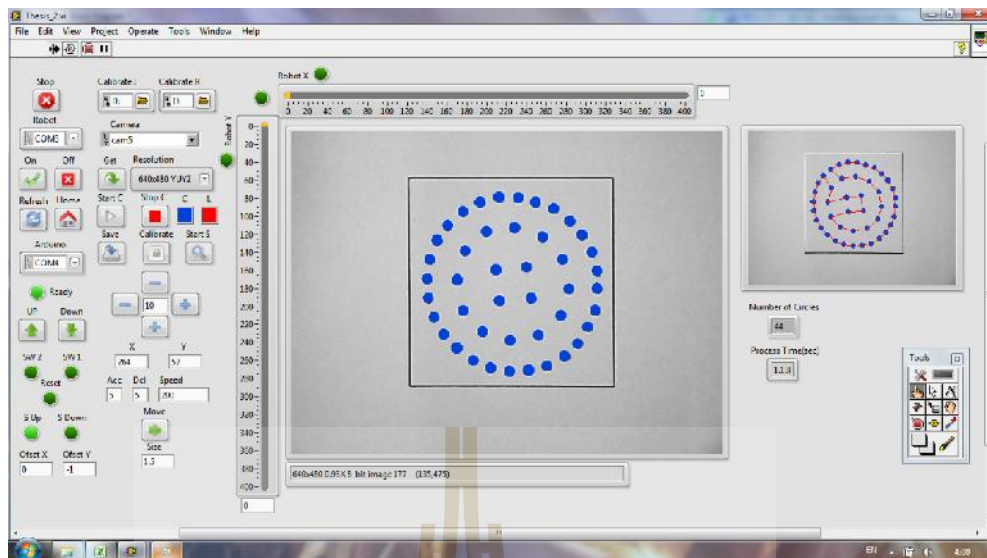


(b)

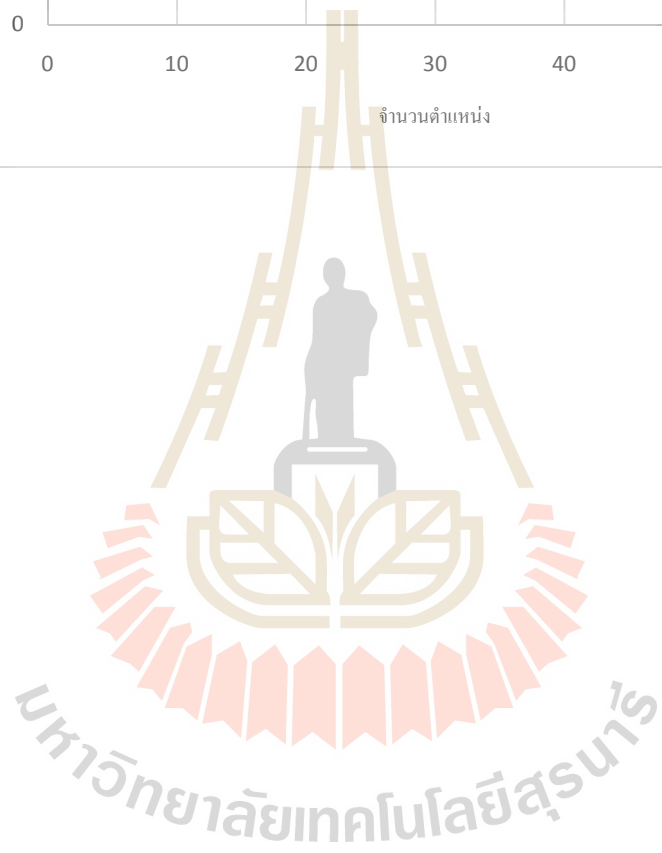
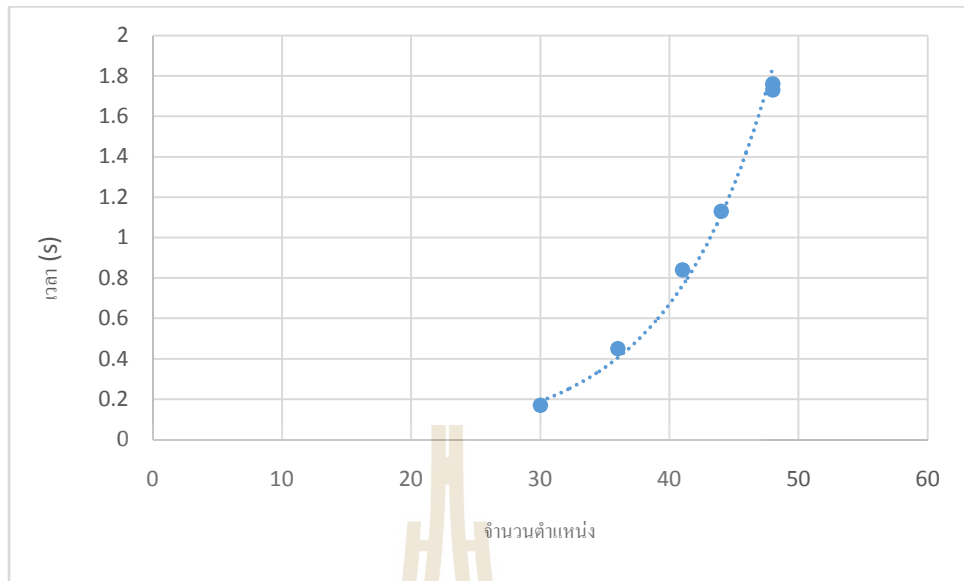




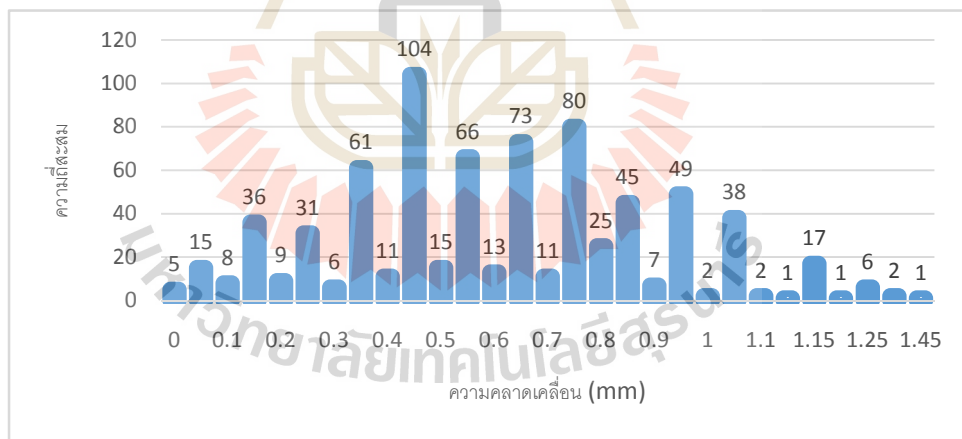
มหาวิทยาลัยเทคโนโลยีสุรนารี

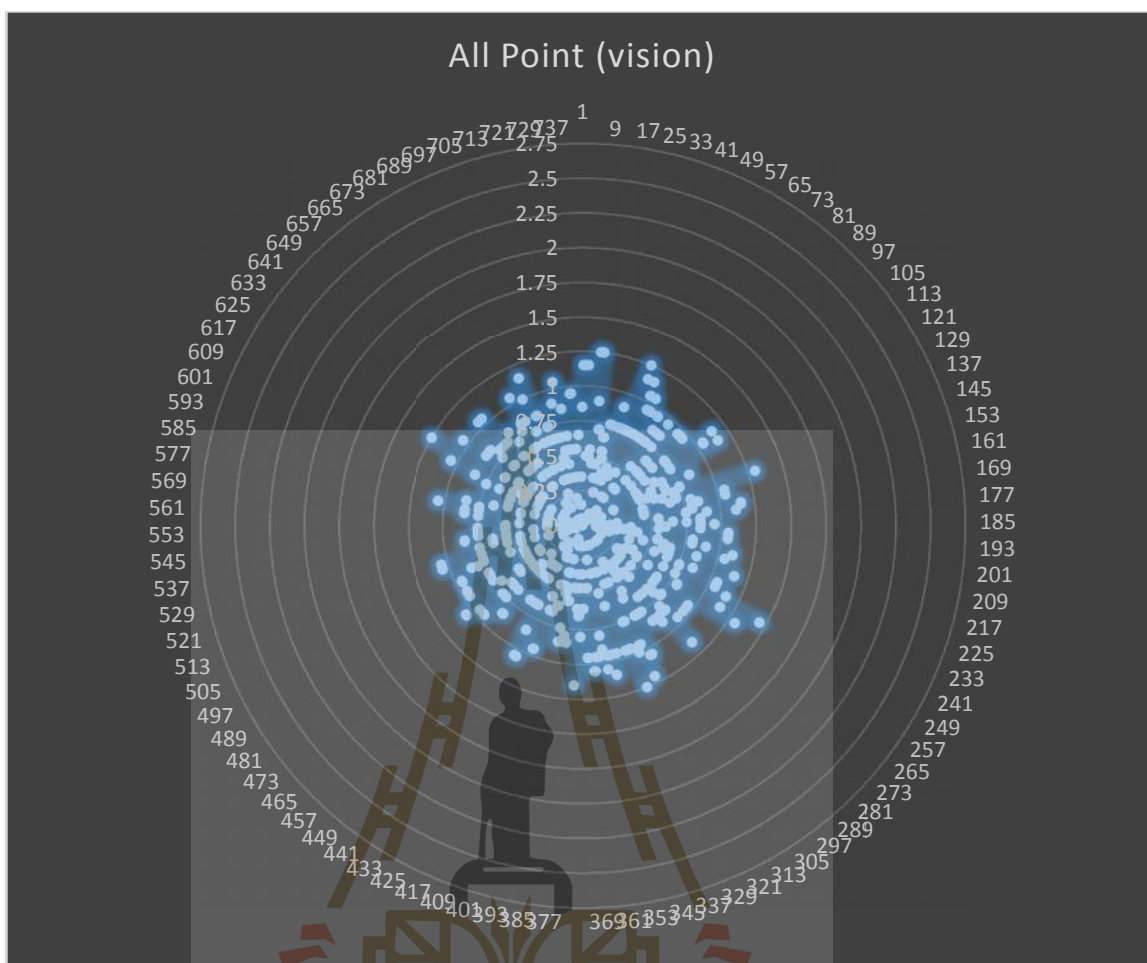


รูปแบบ	จำนวนตำแหน่ง	เวลา (s)
1	48	1.76
2	48	1.73
3	36	0.45
4	44	1.13
5	30	0.17
6	41	0.84



รูปแบบที่	จำนวนตำแหน่ง	ความคลาดเคลื่อน		
		สูงสุด (mm)	ต่ำสุด (mm)	เฉลี่ย (mm)
1	48	1.25	0.05	0.66
2	48	1.45	0.10	0.67
3	36	1.25	0.00	0.64
4	44	1.05	0.05	0.55
5	30	1.05	0.00	0.46
6	41	1.25	0.00	0.55
รวม	741	1.45	0.0	0.59

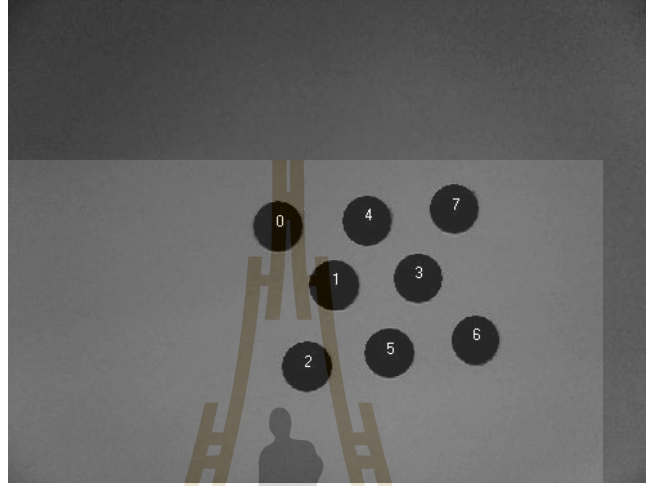




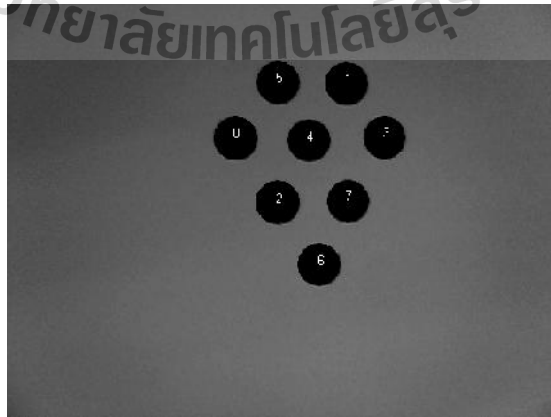
รูปที่ 4.8 ลักษณะความคลาดเคลื่อนทั้ง 741 ตำแหน่ง

#### 4.6 ผลการทดสอบการเปรียบเทียบอัลกอริทึม

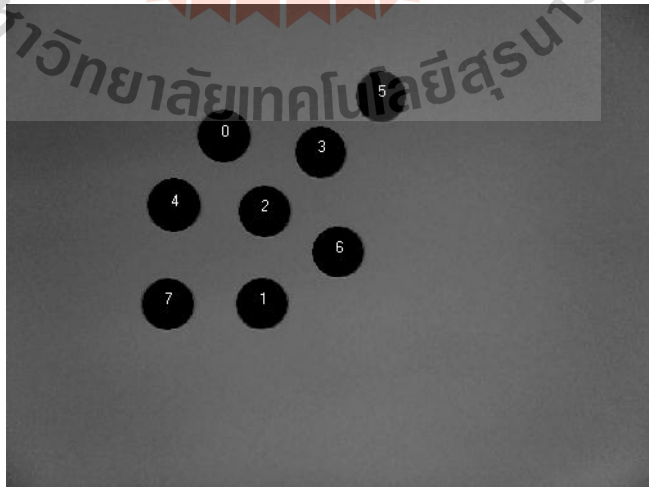
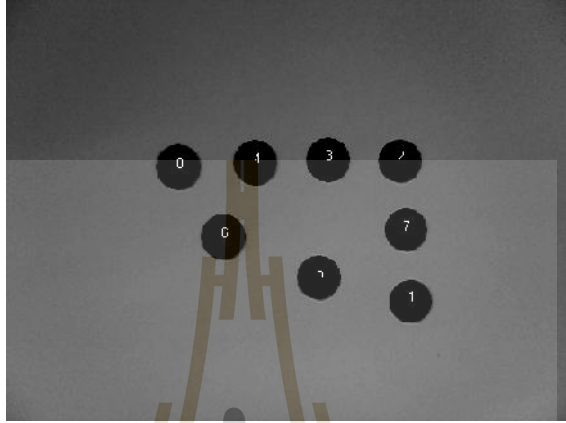
ในการทดสอบในการหาเส้นทางที่สั้นที่สุดผู้วิจัยได้ทำการเขียนโปรแกรมเพื่อทดสอบระหว่างการหาเส้นทางแบบ Brute force กล่าวคือการหาเส้นทางที่เป็นไปได้ทั้งหมดจากนั้นจึงทำการหาเส้นทางที่สั้นที่สุด หลังจากนั้นนำมาเปรียบเทียบกับ การหาเส้นทางที่สั้นที่สุดโดยใช้ Branch and bound ร่วมกับ 2-Opt และในการทดสอบนี้ใช้รูปแบบในการทดสอบทั้งหมด 5 รูปแบบและมีตำแหน่งในการเคลื่อนที่ 8 ตำแหน่ง ซึ่งได้ผลการทดสอบดังนี้



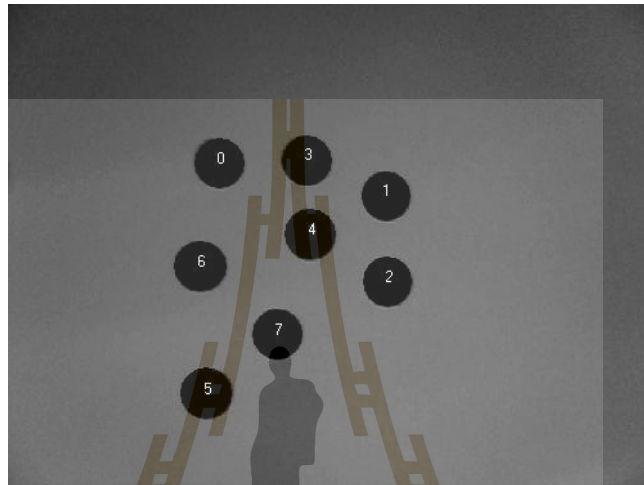
มหาวิทยาลัยเทคโนโลยีสุรนารี







มหาวิทยาลัยเทคโนโลยีสุรนารี



## บทที่ 5

### สรุปผลและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

งานวิจัยนี้เป็นการออกแบบและพัฒนาอัลกอริทึมสำหรับการหาเส้นทางที่สั้นที่สุดเพื่อหาเส้นทางการเคลื่อนที่ของหุ่นยนต์โดยใช้การมองเห็นของเครื่องจักร โดยที่ปัญหาที่ทำการศึกษานี้เป็นการแก้ปัญหาคำถามการเดินทางของพนักงานขาย (TSP) ซึ่งจะประกอบด้วย 2 อัลกอริทึมคือ ใช้ Branch and Bound ในการหาเส้นทางเริ่มต้น และหลังจากนั้นจะเป็นในส่วนของการปรับปรุงระยะทางให้สั้นลงโดยใช้ 2-Opt อัลกอริทึม โดยมีการทดสอบรูปแบบของตำแหน่งสำหรับการเคลื่อนที่ทั้งหมด 6 รูปแบบ และมีจำนวนของตำแหน่งที่แตกต่างกันตั้งแต่ 30 ถึง 48 ตำแหน่ง เพื่อให้เครื่องจักรทำงานได้อย่างมีประสิทธิภาพส่วนที่ต้องคำนึงถึงแบ่งออกเป็น 2 ส่วนดังนี้

##### 5.1.1 อัลกอริทึมสำหรับการหาเส้นทางที่สั้นที่สุด

จากการทดสอบอัลกอริทึมพบว่าอัลกอริทึมสามารถคำนวณหาเส้นทางได้อย่างรวดเร็ว แต่ถึงอย่างไรก็ตามลักษณะการเติบโตของเวลาที่ใช้ในการประมวลผล สำหรับการหาเส้นทางที่สั้นที่สุด ก็ยังเป็นแบบเอ็กซ์โพเนนเชียล เมื่อประมวลผลกับตำแหน่งที่มีจำนวนมากขึ้น และจากการทดสอบพบว่าลักษณะการเรียงตัวของตำแหน่งสำหรับการเคลื่อนที่ก่อนที่มีการประมวลผล มีผลต่อความแน่นอนในการหาเส้นทางด้วยวิธีการของ Branch and Bound อัลกอริทึม เนื่องจากลักษณะการเรียงตัวของตำแหน่งที่ได้จากการประมวลผลของ LabVIEW Vision module นั้นจะมีความแตกต่างกัน ถึงแม้ว่าเราจะใช้รูปแบบของตำแหน่งที่ใช้ในการทดสอบจะเหมือนกัน ส่งผลให้ผลการหาเส้นทางที่สั้นที่สุดนั้นยังไม่มีความแน่นอน ซึ่งในการใช้งานจริงจะสามารถนำไปพัฒนาต่อยอดได้ โดยการปรับปรุงลักษณะการเรียงตัวของตำแหน่ง เพื่อให้ได้เส้นทางที่แน่นอน และการปรับปรุงอัลกอริทึมโดยใช้การประมวลผลแบบ Multitasking หรือการประมวลผลแบบ Parallel ก็สามารถทำให้การประมวลผลเร็วขึ้นได้

##### 5.1.2 การประมวลผลภาพและการควบคุมหุ่นยนต์

จากการทดสอบการควบคุมหุ่นยนต์ด้วยการใช้ระบบการมองเห็นของเครื่องจักรมีความคลาดเคลื่อนสูงสุดอยู่ที่ 1.45 มิลลิเมตร ความคลาดเคลื่อนต่ำสุดอยู่ที่ 0.00 มิลลิเมตร และมีความคลาดเคลื่อนโดยเฉลี่ยอยู่ที่ 0.59 มิลลิเมตร อย่างไรก็ตามความคลาดเคลื่อนที่เกิดขึ้นมีหลายปัจจัยที่เกี่ยวข้องอันได้แก่

- การเกิดความผิดเพี้ยนของภาพและการปรับปรุงความผิดเพี้ยนของภาพ
- อัลกอริทึมที่ใช้ในการหาวงกลมสำหรับการเคลื่อนที่ ซึ่งบางครั้งก็ไม่ได้จุดกึ่งกลางของวงกลม
- ความคลาดเคลื่อนของโรบอท ซึ่งในความเป็นจริงแล้วโรบอทเองก็มีความคลาดเคลื่อนของตัวเอง ซึ่งโรบอทที่เรานำมาใช้มีความคลาดเคลื่อนอยู่ที่ไม่เกิน  $\pm 0.50$  มิลลิเมตร

## 5.2 ข้อเสนอแนะ

5.2.1 เนื่องจากการวิจัยนี้ใช้ระบบการมองเห็นของเครื่องจักรหรือการประมวลผลภาพสิ่งที่ต้องคำนึงถึงสำหรับการประมวลผลได้แก่ แสงจากสิ่งแวดล้อมภายนอกที่จะมีผลกระทบต่อ การประมวลผลภาพ ระยะชัดลึกหรือ Depth of field พื้นที่ที่กล้องสามารถเก็บภาพได้หรือ Field of view ลักษณะของพื้นผิว ประเภทของวัสดุ ความสูงของวัตถุที่สนใจ ชนิดของเลนส์ คุณภาพของกล้องและมุมของแสงที่ตกกระทบ สิ่งเหล่านี้ล้วนมีผล ต่อลักษณะของภาพที่ได้ และขั้นตอนในการประมวลผลภาพทั้งสิ้น ดังในเราจึงต้องความคุมสิ่งเหล่านี้ให้มีสภาพแวดล้อมให้ใกล้เคียงกับพื้นที่ๆเราจะใช้งานจริง หรือการเปลี่ยนแปลงให้น้อยที่สุดเท่าที่เป็นไปได้ เพื่อให้ได้การประมวลผลภาพที่มีประสิทธิภาพมากที่สุด

5.2.2 ในงานวิจัยนี้ใช้วิธีการแก้ปัญหาของพนักงานขายซึ่งเป็นการแก้ปัญหาที่มีความซับซ้อนของปัญหาซึ่งอยู่ในระดับที่ยากหรือ NP-Hard การเลือกอัลกอริทึมที่นำมาใช้แก้ปัญหาต้องมีความเหมาะสม ทั้งในเรื่องของขนาดของข้อมูล เวลาที่ใช้ในการประมวลผล และผลที่ยอมรับได้

5.2.3 ควรเลือกโรบอทและกล้องควบคุมที่สามารถทำงานร่วมกันได้อย่างเหมาะสม เพื่อที่จะทำให้ได้การทำงานที่มีประสิทธิภาพสูงสุด รวมถึงการใช้ความเร็วในการเคลื่อนที่ของโรบอทเนื่องจากสิ่งเหล่านี้ส่งผลต่อความคลาดเคลื่อนของโรบอทโดยตรง

5.2.4 การตรวจสอบส่วนต่างๆ ของโปรแกรมให้ทำงานได้อย่างถูกต้อง มีการตรวจจับเพื่อป้องกันและแสดงความผิดพลาดที่เกิดขึ้น สิ่งเหล่านี้ถือว่าเป็นส่วนหนึ่งที่มีความสำคัญในการพัฒนาโปรแกรม ในการเขียนโปรแกรมขนาดใหญ่หรือโปรแกรมที่มีความซับซ้อนเป็นไปไม่ได้เลยที่จะไม่เกิดความผิดพลาด ซึ่งความผิดพลาดที่เกิดขึ้นนั้นบางครั้งก็คาดเดาผลหรือสาเหตุของการเกิดความผิดพลาดได้ยาก ซึ่งจะส่งผลให้การพัฒนาโปรแกรมเกิดของเราเกิดความล่าช้า ดังนั้นการตรวจจับความผิดพลาดของโปรแกรมจึงเป็นสิ่งที่ไม่ควรละเลย

5.2.5 หากต้องการนำอัลกอริทึมไปใช้งานจริงควรเลือกการตัดและเชื่อมเส้นทางใหม่เป็นแบบสุ่มสำหรับอัลกอริทึมของ 2-Opt

## รายการอ้างอิง

- กนต์ธร ชำนิประศาสน์ (2548) **การวัดเชิงกลด้วย LabVIEW**. สาขาวิชาวิศวกรรมเครื่องกล สำนัก  
วิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี 240 หน้า.
- ปกรณ แก้วตระกูลพงษ์ (2559) **การประยุกต์อัลกอริทึมแมชชีนวิชันโดยใช้ NI Vision**, โรงพิมพ์แห่ง  
จุฬาลงกรณ์มหาวิทยาลัย:สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, หน้า 121-151.
- บุญเจริญ ศิริเนาวกุล (2551) **Artificial Intelligence ปัญญาประดิษฐ์ บทที่ 3 การค้นหาแบบฮิวริ  
สติก**, สำนักพิมพ์ท้อป หน้า 6 - 8.
- Richard Wiener. (2003) **Branch and Bound implementation for the travelling salesman  
problem**  
**part1**. Journal of object technology, vol2, no.2, pp 65 - 86
- Richard Wiener. (2003) **Branch and Bound implementation for the travelling salesman  
problem**  
**part1**. Journal of object technology, vol2, no.3, pp. 65 – 76
- Michael Hashsler, Kurt Hornik. (2007) **TSP – Infrastructure for the Travelling salesman  
problem**.  
Journal of Statistical Software, vol.23, issue2, December.
- Justin Jackson, Anouck Girard, Steven Rasmussen, Corey Schumacher, (2010) **A Combined  
tabu  
search and 2-opt heuristic for multiple vehicle routing**. American control conference  
Marriott Waterfront,
- Handa A. Darwish, Ihab Talkhan, (2014) **Reduce complexity divide and conquer algorithm for  
large  
scale TSPs**. International journal of advance science and application, vol.5, no.1
- Kim, S.M., Lee, S.C., and Lee, Y.C. (2006) **Vision Based Automatic Inspection System for  
Nuts  
Welded on the Support Hinge**. *SICE-ICASE International Joint Conference 2006*.

Prommarak, N., and Chamniprasart, K. (2010) **Damage Screw Inspection Computer Vision**

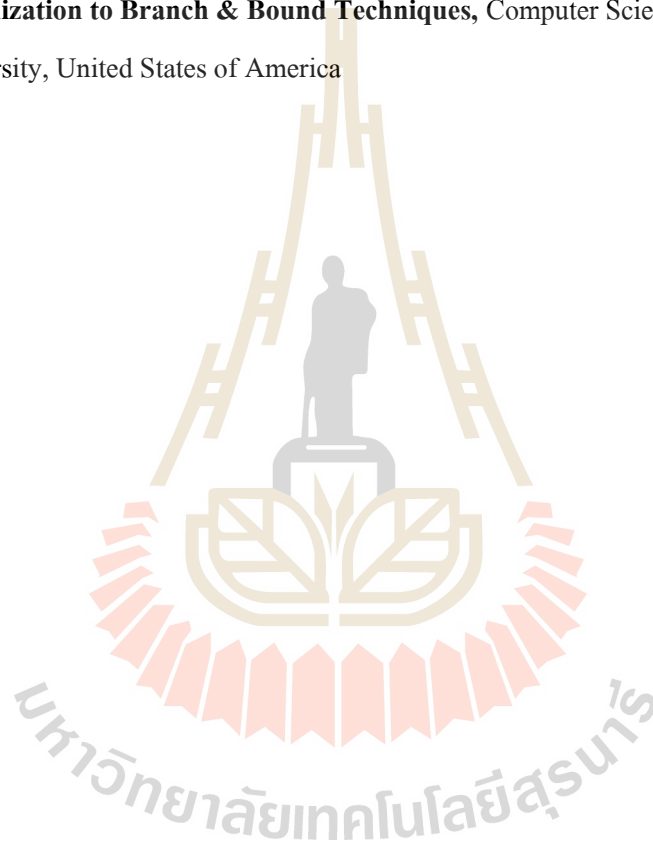
**Prototype Machine.** Proceedings of International Conference on Data Storage Technology

(DST-CON) Bangkok, Thailand .

Hitokazu Matsushita, Ogden Mills, Nathan Lambson, **Adapting 2-Opt And 3-Opt Local**

**Optimization to Branch & Bound Techniques,** Computer Science, Brigham Young

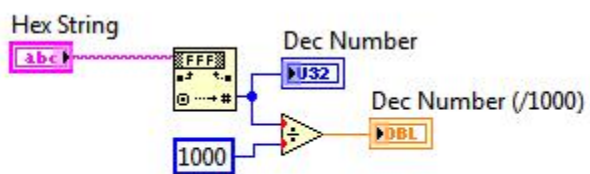
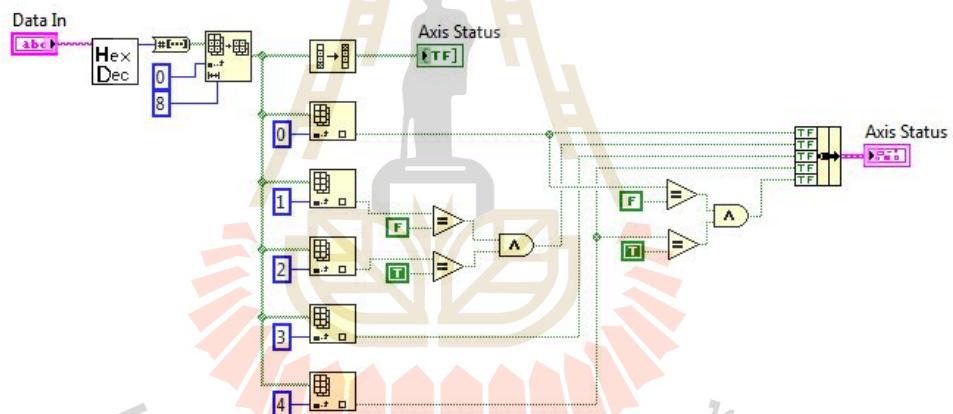
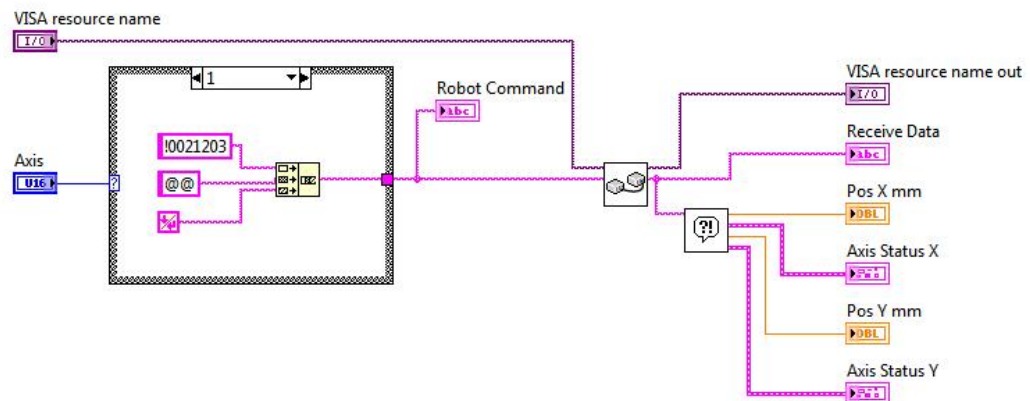
University, United States of America



ภาคผนวก ก

รายละเอียดโปรแกรมย่อย Sub VI ที่ใช้ในงานวิจัย

มหาวิทยาลัยเทคโนโลยีสุรนารี

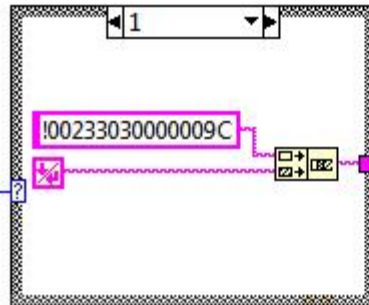




VISA resource name

I/O

Ring  
U16



VISA resource name out

I/O

Receive Data

abc

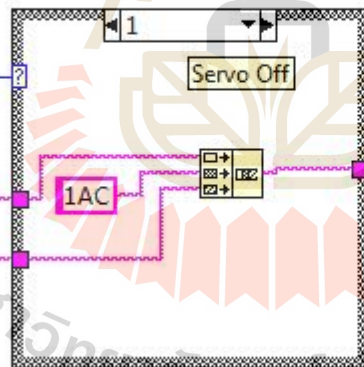
Home Command

abc

VISA resource name

I/O

Ring  
I8



VISA resource name out

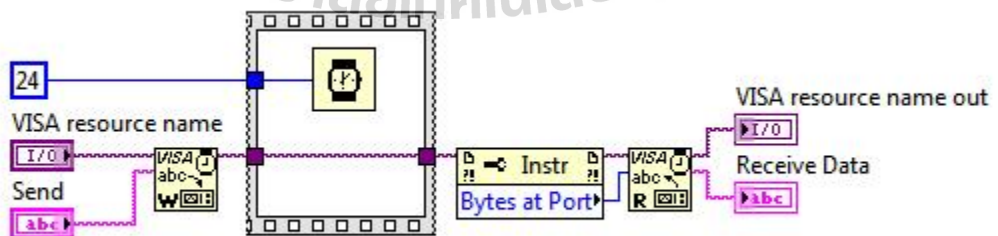
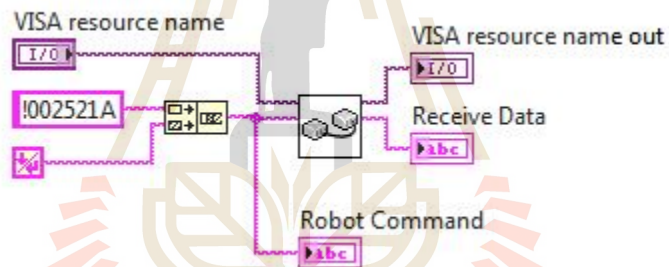
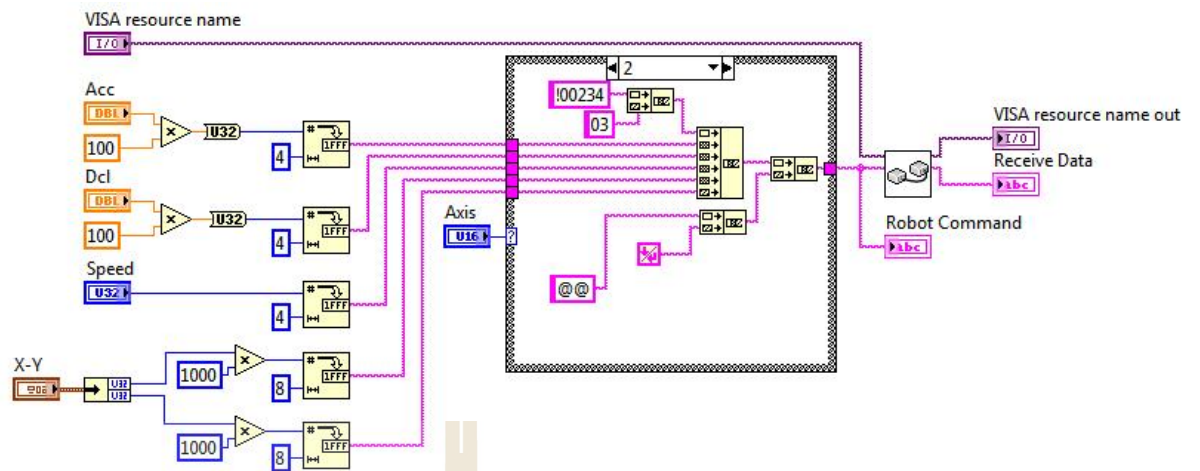
I/O

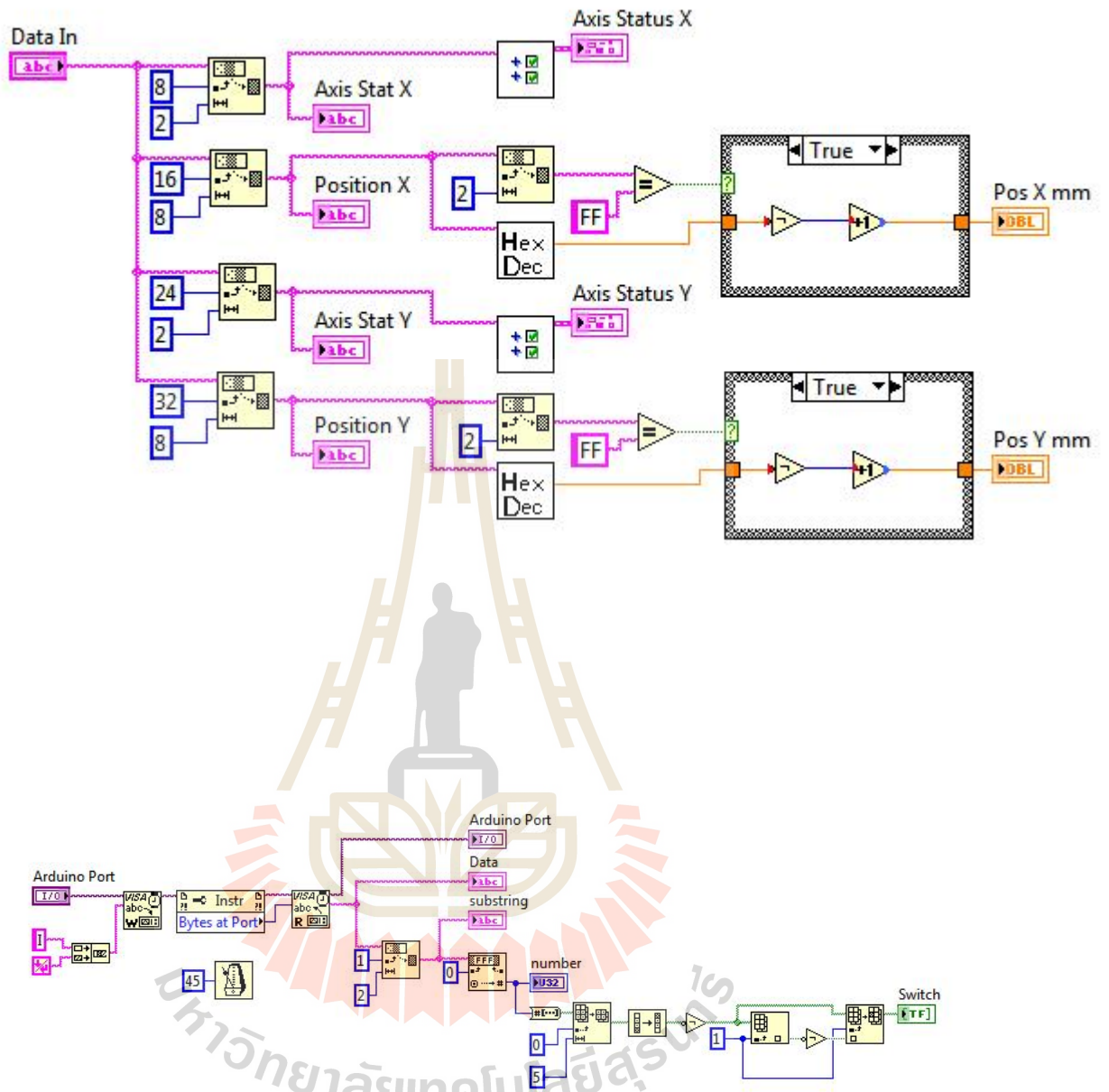
Receive Data

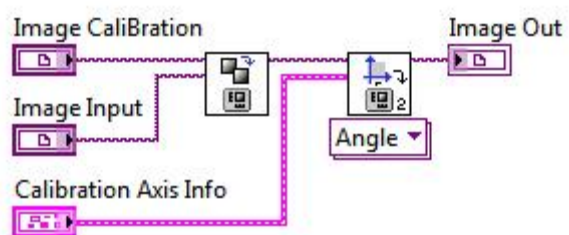
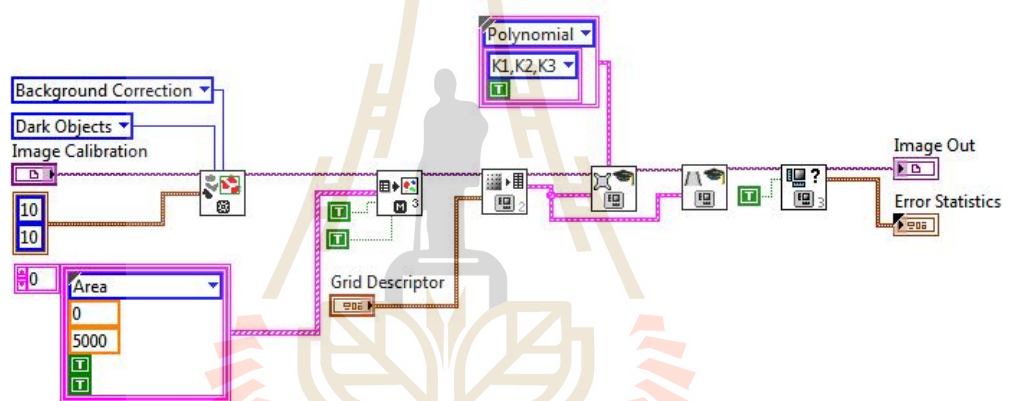
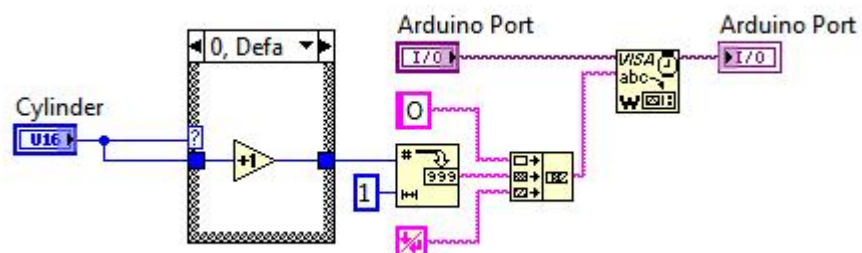
abc

Comand

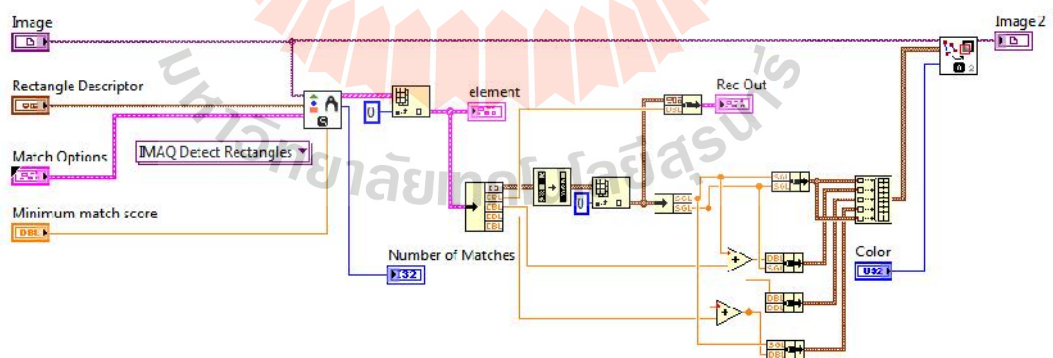
abc

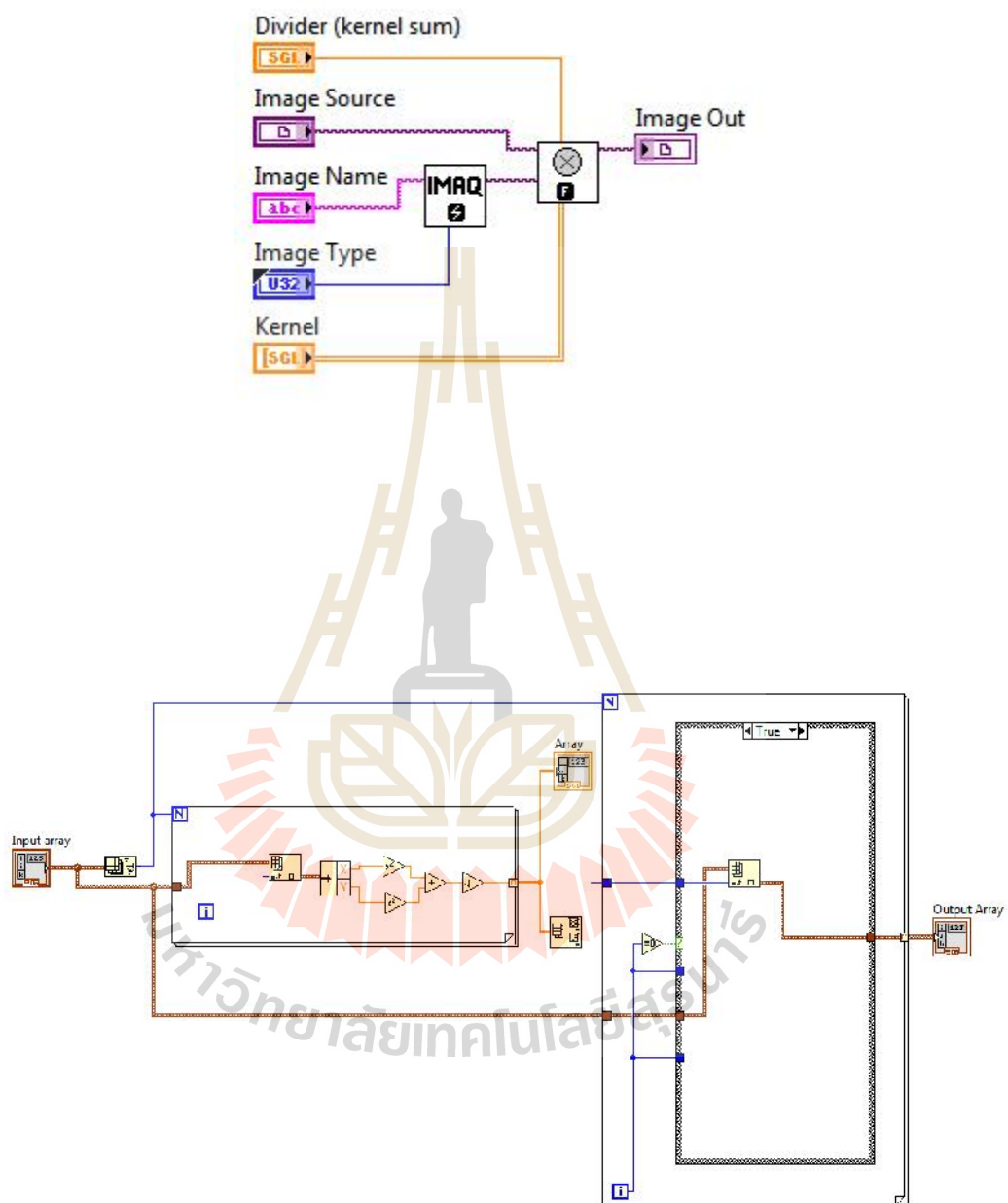




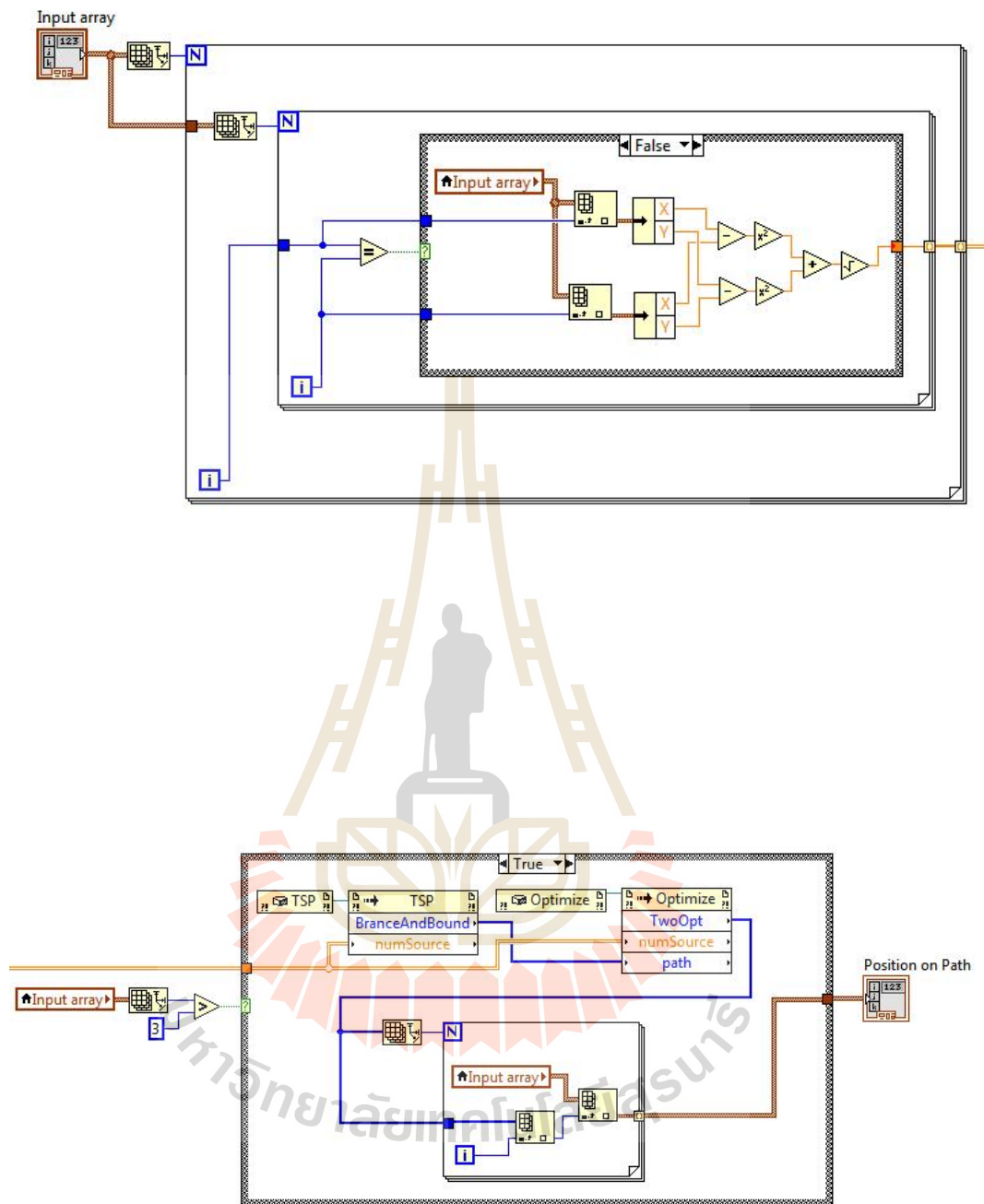


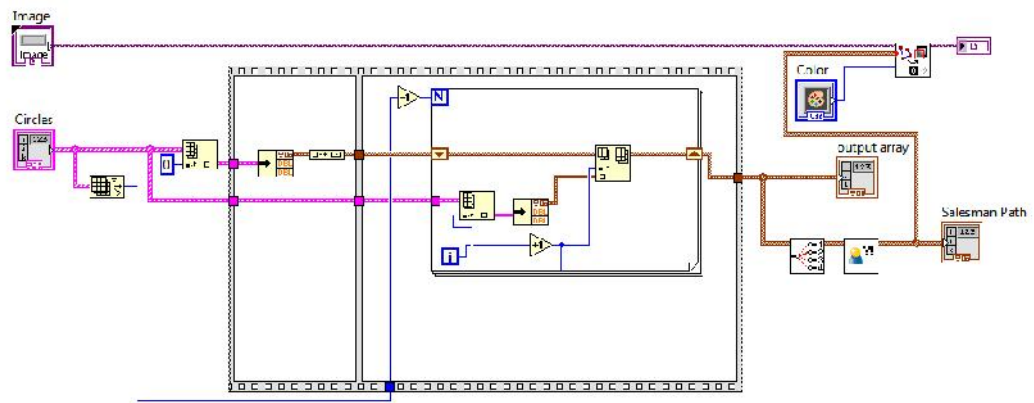
มหาวิทยาลัยเทคโนโลยีสุรนารี













## ประวัติผู้เขียน

นายอริยวัตร วริณฺณ เกิดเมื่อวันที่ 9 กรกฎาคม 2528 เริ่มการศึกษาระดับประถมศึกษาปีที่ 1- 6 ที่โรงเรียนอนุบาลร้อยเอ็ด และมัธยมศึกษาปีที่ 1 – 6 ที่โรงเรียนร้อยเอ็ดวิทยาลัย อำเภอเมือง จังหวัดร้อยเอ็ด สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อ พ.ศ. 2554 และเข้าศึกษาต่อในระดับ วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมการผลิต มหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อ พ.ศ. 2555 ขณะศึกษาต่อในระดับมหาบัณฑิตนั้น ได้เป็นผู้สอนรายวิชาปฏิบัติการของสาขา วิศวกรรมเครื่องกลและสาขาวิชาวิศวกรรมการผลิตได้แก่

1. ปฏิบัติการวิศวกรรมเครื่องกล
2. ปฏิบัติการวิศวกรรมอากาศยาน
3. ปฏิบัติการวิศวกรรมการผลิต
4. ปฏิบัติการวิศวกรรมแมคคาทรอนิกส์
5. ปฏิบัติการระบบอัตโนมัติในอุตสาหกรรม

อีกทั้งเป็นที่ปรึกษาโปรเจกต์นักศึกษาวิศวกรรมการผลิตในโปรเจกต์การควบคุมการเปิดปิด แอร์และไฟ(บางส่วน)ในสำนักวิชาวิศวกรรมเครื่องกลโดยใช้โปรแกรม LabVIEW และยังเป็น ผู้ช่วยวิทยากรอบรมพนักงานบริษัท ซีเกท เทคโนโลยี (ประเทศไทย) จำกัด จังหวัด นครราชสีมา ในหลักสูตร Microcontroller